生物信息学系列教程

空间生态位与细胞通讯重编程

肿瘤微环境空间转录组分析实战教程

Spatial Niche & Communication Reprogramming

in the Tumor Microenvironment

10x Visium • 生态位识别 • 细胞通讯 • 新手实战

第 1 版　2026 年

目录

第 00 章导读与学习路线 **8**

本章目标 8

0.1 这本书讲什么 8

0.2 为什么选这个课题 9

0.3 学习路线图 9

0.4 学习进度表 11

0.5 前置要求与心态 12

0.6 如何使用本书 12

0.7 本书配套资源 13

本章小结 13

练习与思考 14

第 01 章肿瘤微环境与细胞通讯的生物学基础 **15**

本章目标 15

1.1 肿瘤微环境：癌细胞不是孤军奋战 15

1.2 细胞通讯的基本形式：配体与受体 16

1.3 六条关键通讯通路 16

1.4 肿瘤如何“重编程”微环境 18

1.5 为什么“空间”对理解 TME 重要 19

本章小结 19

练习与思考 20

第 02 章空间转录组技术原理：从单细胞到 10x Visium **21**

本章目标 21

2.1 单细胞测序的成就与盲区 21

2.2 空间转录组技术谱系 22

2.3 10x Visium 原理详解 23

2.4 Visium 数据文件结构 24

2.5 空间转录组的三个局限 25

本章小结 25

练习与思考 25

第 03 章生信入门必备：R/Python 与核心数据结构 **27**
本章目标 . . . 27
3.1 R 语言快速上手 . . . 27
3.2 R 绘图理念：ggplot2 . . . 28
3.3 Python 快速上手 . . . 29
3.4 两个生态的核心对象 . . . 29
3.5 常用文件格式 . . . 30
3.6 20 分钟热身练习 . . . 31
3.7 推荐学习资源 . . . 32
本章小结 . . . 32
练习与思考 . . . 33

第 04 章空间生态位与通讯推断方法学总览 **34**
本章目标 . . . 34
4.1 三个核心概念：生态位、空间域与邻域 . . . 34
4.2 生态位识别方法谱系 . . . 36
4.3 细胞通讯推断方法谱系 . . . 38
4.4 方法选择决策树：本书推荐组合 . . . 39
4.5 两张图读懂本书的生态位视角 . . . 40
4.6 为什么“先分生态位、再做通讯”更有生物学意义 . . . 41
本章小结 . . . 41
练习与思考 . . . 41

第 05 章课题设计：创新点拆解与实验路线 **42**
本章目标 . . . 42
5.1 一句话课题模板 . . . 42
5.2 创新点拆解：三个层次 . . . 43
5.3 研究假设：可证伪的才是假设 . . . 43
5.4 分析路线图：从数据到论文 . . . 44
5.5 预期图表清单：先想好结果长什么样 . . . 47
5.6 课题常见误区 . . . 47
5.7 练习：写出你的课题 . . . 48
本章小结 . . . 48
练习与思考 . . . 48

第 06 章环境搭建与工具链 **50**
本章目标 . . . 50
6.1 为什么用 conda/mamba 管理环境 . . . 50
6.2 创建 Python 空间转录组环境 . . . 51
6.3 安装 R 环境与关键包 . . . 51

6.4 Windows 用户必看：Rtools 与依赖 53
6.5 环境导出与复现 53
6.6 验证安装：最小可运行测试 54
6.7 常见安装坑与解法 55
6.8 配套脚本说明 56
本章小结 56
练习与思考 57

第 07 章数据获取：Visium 公开数据集 58
本章目标 58
7.1 数据集选型：为什么是人乳腺癌 demo 58
7.2 10x 官网下载流程 59
7.3 备选数据源 60
7.4 数据清单与校验（data-inventory） 61
7.5 目录组织建议 61
7.6 下载脚本：code/02_download.R 核心部分 62
7.7 练习：下载并登记 63
本章小结 63
练习与思考 64

第 08 章预处理与质控 65
本章目标 65
8.1 读入数据：Load10X_Spatial 65
8.2 QC 指标的含义：先看懂三个数字 66
8.3 阈值设定：先看分布，不拍脑袋 67
8.4 归一化：LogNormalize 还是 SCTransform？ 68
8.5 高变基因、PCA 与聚类 69
8.6 空间可视化入门 70
8.7 结果解读：QC 图与聚类图 70
8.8 动手练习：调整阈值，观察 spot 数量变化 71
本章小结 72
练习与思考 72

第 09 章降维聚类与细胞类型注释 73
本章目标 73
9.1 为什么 spot 需要注释：混合信号的真相 73
9.2 策略 A：marker 基因直接注释（最轻量，先做它） 74
9.3 策略 B：单细胞参考去卷积（更定量） 75
9.4 策略 C：与公共单细胞图谱整合注释 76
9.5 结果解读：注释结果图 77

9.6 常见问题排查 . . . 78
9.7 动手练习 . . . 78
本章小结 . . . 79
练习与思考 . . . 79

第 10 章空间生态位识别 80
本章目标 . . . 80
10.1 三步法总览 . . . 80
10.2 第一步：空间域分割（BayesSpace 实操） . . . 81
10.3 第二步：邻域富集验证（Squidpy 实操） . . . 82
10.4 第三步：生态位命名与特征化 . . . 83
10.5 结果解读：生态位图与邻域富集热图 . . . 84
10.6 生态位数量的生物学判断 . . . 87
10.7 动手练习：跑 BayesSpace，比较 q = 4 / 6 / 8 . . . 87
本章小结 . . . 88
练习与思考 . . . 88

第 11 章细胞通讯分析基础：CellChat 89
本章目标 . . . 89
1 为什么推断细胞通讯：表达≠通讯 . . . 89
2 CellChat 原理（讲人话） . . . 90
3 安装 CellChat v2 与对象构建 . . . 91
4 完整分析流程 . . . 92
5 经典可视化逐个讲解 . . . 93
6 结果解读示例：读图说话 . . . 94
7 注意事项 . . . 95
本章小结 . . . 95
练习与思考 . . . 95

第 12 章生态位内细胞通讯分析 96
本章目标 . . . 96
1 核心思路：为什么要按生态位拆开分析 . . . 96
2 实操：按生态位拆分并独立建 CellChat . . . 97
3 生态位内通讯特征解读示例 . . . 99
4 生态位内 vs 全局通讯：为什么分区能发现新信号 . . . 99
5 结果整理：各生态位通讯网络汇总表 . . . 100
本章小结 . . . 101
练习与思考 . . . 101

第 13 章通讯重编程：生态位间差异分析 102

本章目标 . . . 102
1 什么是 “通讯重编程”：操作化定义 . . . 102
2 差异通讯数量/强度比较 . . . 103
3 差异信号通路识别 . . . 104
4 差异配体-受体对分析：谁在具体地 “多说/少说” . . . 105
5 空间活性验证：三重证据闭环 . . . 106
6 重编程故事化：Sankey 流量图 . . . 108
7 结果解读示例：一段示范性 Results 文字 . . . 109
8 注意事项 . . . 109
本章小结 . . . 110
练习与思考 . . . 110

第 14 章论文级图表与可视化 111
本章目标 . . . 111
14.1 论文级图表通用规范 . . . 111
14.2 关键图表制作清单 . . . 113
14.3 一图一故事：结果图编排策略 . . . 116
14.4 中文字体设置 . . . 118
14.5 常见图表错误与修正 . . . 119
本章小结 . . . 119
练习与思考 . . . 120

第 15 章结果解读与论文写作 121
本章目标 . . . 121
15.1 如何讲 “重编程” 故事 . . . 121
15.2 结果章节四段式 . . . 122
15.3 图表-文字对应与数字一致性检查 . . . 123
15.4 方法章节写作：可复现是硬标准 . . . 124
15.5 投稿期刊建议 . . . 124
15.6 预印本 bioRxiv 流程 . . . 125
15.7 写作自查清单 . . . 125
本章小结 . . . 126
练习与思考 . . . 126

第 16 章常见问题与排错 127
本章目标 . . . 127
16.1 安装类问题 . . . 127
16.2 运行类问题 . . . 129
16.3 可视化类问题 . . . 132
16.4 结果类问题 . . . 133

16.5 交付前检查清单 135
本章小结 135
练习与思考 135

附录 A 术语表 **137**
A–D 137
E–L 138
M–R 139
P–S 139
S–Z 140

附录 B 资源与参考文献 **142**
B.1 工具官网与文档清单 142
B.2 新手友好教程资源 143
B.3 参考文献 144
B.4 数据资源 145

附录 C 示例数据与代码清单 **146**
C.1 配套代码清单 146
C.2 示例数据说明 147
C.3 复现全书的最低路径 148

第 00 章导读与学习路线

本章是全书的总入口。读完后，你会知道这本书要带你完成一件什么事、为什么要做这件事、按什么节奏学、需要什么准备。

本章目标

读完本章后，你应该能：

1. 用自己的话说清楚“空间生态位 × 细胞通讯重编程”课题的全貌；
2. 说出传统单细胞测序与空间转录组的核心差异，以及本书创新点所在；
3. 对照学习路线图（图 0-1）指出全书 16 章分属哪个阶段；
4. 根据三周/六周进度表制定自己的学习计划；
5. 明确本书配套资源（17 张图、10 个 R 脚本、示例数据）的位置与用途。

0.1 这本书讲什么

先给一个“全景快照”。

肿瘤不是一堆癌细胞的简单堆积，而是一个“微型社会”：里面有癌细胞、成纤维细胞、免疫细胞、血管内皮细胞，还有把它们黏在一起的**细胞外基质**（extracellular matrix, ECM）。这些细胞之间不断通过**配体-受体**（ligand-receptor）对话——有的促进生长，有的抑制免疫，有的拉来更多血管。我们把这套对话叫作**细胞通讯**（cell-cell communication, CCC）。

本书要带你完成的课题，用一句话说是：

> **比较肿瘤组织不同空间生态位的细胞通讯网络，揭示微环境中的“通讯重编程”信号。**

拆开来看，这个课题分三步。

第一步：识别空间生态位（spatial niche）。把一块组织切片划分成功能不同的区域——**肿瘤核心**（tumor core）、**侵袭前沿**（invasive front）、**免疫浸润区**（immune-infiltrated region）、**基质区**（stromal region）。这四个区域细胞组成不同、行为不同、通讯也不同。

第二步：比较各生态位的细胞通讯网络。在每个生态位内部，分别构建“谁向谁发了什么信号”

的通讯网络，统计通讯的数量、强度与通路，然后横向比较：侵袭前沿和肿瘤核心的通讯有什么不一样？免疫浸润区里 T 细胞和 B 细胞在聊什么？

第三步：发现“通讯重编程”信号。所谓**通讯重编程**（communication reprogramming），指的是肿瘤组织不同区域的通讯网络发生了系统性改变——比如免疫抑制信号（PD-L1/PD-1）只在某个生态位高活性，侵袭相关通路（TGF-β、CXCL12）在侵袭前沿被“打开”。这种“位置依赖的通讯差异”，正是我们要找的核心发现。

现在看不懂 PD-L1/PD-1、TGF-β 这些名词没关系，第 01 章会逐个讲清楚。

0.2 为什么选这个课题

三个理由。

理由一：传统单细胞测序丢掉了空间位置。单细胞 RNA 测序（single-cell RNA sequencing, scRNA-seq）把组织解离成一个个细胞，能给出“每个细胞表达什么”，却给不出“这个细胞原来在组织的哪个位置”。而肿瘤的很多关键行为恰恰是位置决定的：肿瘤核心缺氧、侵袭前沿与基质交界、免疫浸润区聚集大量免疫细胞——这些信息在解离的那一刻就永久丢失了。

理由二：空间转录组把位置找回来了。空间转录组（spatial transcriptomics）在组织切片上直接测表达，每个测量点都带着坐标。于是我们第一次能回答：“这片组织里，不同位置的细胞分别是谁、在干什么、怎么交流。”

理由三：创新点在“生态位视角”。多数空间转录组分析做的是“全局分析”：把整张切片当成一个整体，算平均通讯。但肿瘤恰恰是高度异质的——全局平均会掩盖位置特异的信号。本书的卖点是把分析单位从“整张切片”下沉到“生态位”，在生态位内部算通讯、生态位之间比差异。这个视角本身就是可发表的创新点（第 05 章会专门拆解）。

0.3 学习路线图

全书 16 章加 3 个附录，按四阶段组织：

- **基础铺垫（第 01–04 章）**：生物学基础、空间转录组原理、R/Python 入门、生态位与通讯方法学总览；
- **课题设计（第 05 章）**：把“生态位 × 通讯重编程”拆成可执行的假设与步骤；
- **分析主线（第 06–13 章）**：环境搭建→数据获取→预处理质控→聚类注释→生态位识别→通讯分析→差异通讯（重编程）；
- **写作交付（第 14–16 章）**：论文级图表、结果解读与写作、常见问题排错。

动手之前，先确认环境能不能跑本书的代码：

图 1: 图 0-1 全书技术路线图。四阶段彩色流程图：基础铺垫→课题设计→分析主线（环境数据→预处理→注释→生态位→通讯→重编程）→写作交付。建议把这张图贴在桌前，随时确认自己走到哪一步。

```r
# 本书技术栈速查（包安装详见第 06 章，这里先确认 R 版本）
R.version.string                    # 本书要求 R ≥ 4.3

if (requireNamespace("Seurat", quietly = TRUE)) {
  packageVersion("Seurat")          # 本书要求 Seurat v5
} else {
  message("Seurat 还没装——别急，第 06 章会手把手教你。")
}
```

运行结果解读：第一行输出你的 R 版本，≥ 4.3 即可；如果 Seurat 未安装，你会看到一句提示而不是红色报错，这是正常现象。环境搭建的完整流程在第 06 章。

0.4 学习进度表

你可以选“三周速成版”或“六周从容版”。三周版适合每天能投入 2–3 小时、有基础编程经验的人；六周版适合每天 1 小时、边学边练的新手。两个版本都建议按“通读一章→复现代码→完成练习”的节奏推进，不要只读不练。

三周速成版

周次	内容	产出
第 1 周	第 00–03 章：导读、生物学基础、技术原理、生信入门	环境可用；跑通第 03 章的 20 分钟热身练习
第 2 周	第 04–07 章：方法学总览、课题设计、环境搭建、数据下载	写出自己的“一句话课题”；Visium 数据下载完成
第 3 周	第 08–13 章：预处理→注释→生态位→通讯→重编程	完整分析流程跑通，得到差异通讯与重编程信号
之后	第 14–16 章	论文级图表与写作初稿

六周从容版

周次	内容	产出
第 1 周	第 00–01 章	理解全书框架与肿瘤微环境概念
第 2 周	第 02–03 章	掌握 Visium 原理；完成 R/Python 基础练习
第 3 周	第 04–06 章	方法学总览；课题一句话；环境搭建完成

周次	内容	产出
第 4 周	第 07–08 章	数据下载完成；预处理与 QC 图（对应 fig07_qc.png）
第 5 周	第 09–10 章	细胞类型注释与生态位识别图（对应 fig09、fig10）
第 6 周	第 11–13 章	分区通讯与重编程分析（对应 fig12–fig16）
之后	第 14–16 章	论文级图表、写作与排错

0.5 前置要求与心态

硬件上，一台 8–16 GB 内存的电脑即可——Visium 单切片在 R 中约占 2–6 GB 内存（见 00-TECH 第 2 节）；需要安装 R ⩾ 4.3、RStudio 与 Python ⩾ 3.10，具体流程在第 06 章。

知识上，本书默认读者 “会一点点 R 或 Python”，但第 03 章会补上够用的基础，所以零基础也能跟。你不需要是编程高手，也不需要背下所有生物学名词——名词会反复出现，见多了自然记住。

心态上，请带上三样东西：

1. **耐心**。报错是常态，不是你的错。每一行报错都是学习机会，本书第 16 章专门讲排错。
2. **复现精神**。先老老实实跑通本书代码，再改参数、换数据、加自己的想法。复现是创新的前提。
3. **质疑习惯**。看到结论先问 “这是分析结果还是生物学解释？”——本书会反复提醒：通讯推断是**计算预测**，不是直接测量。

0.6 如何使用本书

每章都按固定结构组织（本章目标→正文→代码→图→小结→练习与思考），建议按四步使用：

1. **通读一章**，只看文字和图，建立整体印象，不急着跑代码；
2. **复现代码**，把代码块逐个跑通，看 “运行结果解读”，理解每段代码在干什么；
3. **对照图注**，把跑出的结果和书里的图（figures/ 目录）对照，确认自己读懂了；
4. **做练习**，完成每章末尾的 “练习与思考”，尤其是 “动手题”——那是检验你真正学会的试金石。

两个提醒：代码里的参数（比如聚类分辨率、过滤阈值）不是圣经，第 08 章会教你 “基于分布而非拍脑袋” 地设定阈值；每章末尾的参考文献按 [作者年份] 格式标注，附录 B 汇总，需要深挖时按图索骥。

0.7 本书配套资源

资源	数量	位置与说明
教学图	17 张	figures/ 目录；fig01–fig06 为手绘概念示意图，fig07–fig17 为 “模拟数据演示” 的分析结果图
可运行脚本	10 个 R 脚本	code/01_setup.R 到 code/10_summary.R，对应第 06–15 章，按编号顺序运行
示例数据	主示例 + 模拟数据	主示例为 10x Visium 人乳腺癌 demo（官网公开下载，第 07 章）；教学图为模拟数据风格

配套目录长这样（数据下载前先建好骨架）：

```
# 本书配套目录一览（在项目根目录运行；Windows 用 dir 代替 ls）
ls spatial-niche-tutorial/
# tutorial/  各章 markdown 源文件
# figures/   17 张教学图
# code/      10 个可运行 R 脚本
# data/      下载的数据（第 07 章创建）
# envs/      环境文件（第 06 章创建）
```

运行结果解读：输出列出项目根目录下的文件夹。data/ 与 envs/ 要到第 06、07 章才会创建，现在没有是正常的。

需要说明的是：fig07–fig17 是**模拟数据演示**，目的是让你看到 “真实分析会长什么样”；你的任务是跟着第 07–13 章，用真实数据自己跑出同款图。

本章小结

- 本书目标：完成 “识别生态位→比较生态位通讯→发现通讯重编程” 的完整课题；
- 选题理由：单细胞丢位置、空间转录组保位置、创新点在生态位视角；
- 学习节奏：16 章四阶段，三周速成或六周从容皆可；
- 心态：耐心、复现、质疑；前置要求很低，坚持要求很高。

练习与思考

1.（动脑题）用一句话向一位不懂生信的朋友解释：为什么 “细胞在组织里的位置” 很重要？
2.（动脑题）对照图 0-1，指出 “细胞类型注释”（第 09 章）属于哪个阶段？它和 “生态位识别”（第 10 章）是什么关系？
3.（动手题）运行 0.3 节的 R 代码，记录你的 R 版本；如果 Seurat 未安装，记下提示信息，第 06 章会解决它。
4.（动手题）浏览 figures/ 与 code/ 目录，把 17 张图按 “示意图/分析演示图” 分成两类，对照 00-FIGURES 清单核对。
5.（思考题）本书反复强调 “通讯推断是计算预测”。你觉得在什么情况下，计算预测的结果需要实验验证？

第 01 章肿瘤微环境与细胞通讯的生物学基础

本章是全书生物学知识的 “地基”。没有编程，只有概念——但请认真读，后面每一章都会用到这里的名词。

本章目标

读完本章后，你应该能：

1. 说出肿瘤微环境的六大组分及其各自角色；
2. 区分自分泌、旁分泌、近分泌、内分泌四种通讯形式；
3. 说出六条教学通讯通路的配体、受体与生物学含义；
4. 用 “免疫抑制、血管生成、EMT、促纤维化” 四条主线解释肿瘤如何重编程微环境；
5. 理解 “空间” 视角对研究肿瘤微环境为什么必不可少。

1.1 肿瘤微环境：癌细胞不是孤军奋战

长期以来的 “肿瘤等于癌细胞” 观念已被修正。现在的共识是：肿瘤是一个由多种细胞与细胞外基质共同组成的生态系统，这个系统叫作**肿瘤微环境**（tumor microenvironment, TME）。六个核心成员：

1. **癌细胞**（cancer cell）：肿瘤的主体，携带驱动突变，无限增殖、逃避凋亡；
2. **癌症相关成纤维细胞**（cancer-associated fibroblast, CAF）：被肿瘤 “招募” 并激活的成纤维细胞，分泌细胞外基质、生长因子与趋化因子，是基质区的核心成员；
3. **免疫细胞**：包括 **CD8 T 细胞**（细胞毒性 T 细胞，负责杀伤肿瘤细胞）、**巨噬细胞**（macrophage，常被肿瘤 “策反” 为促肿瘤表型）、**B 细胞**（产生抗体、呈递抗原）、**自然杀伤细胞**（natural killer cell, NK 细胞，天然免疫杀伤）；
4. **血管内皮细胞**（endothelial cell）：构成血管内壁，在肿瘤中大量新生（血管生成）；
5. **细胞外基质**（extracellular matrix, ECM）：胶原（collagen）、纤连蛋白（fibronectin）等构成的支架，既是物理支撑，也储存信号分子；
6. 其他成员：中性粒细胞、肥大细胞、脂肪细胞等，本书从简处理。

这些细胞不是各干各的，而是通过**细胞通讯**（cell-cell communication, CCC）密切对话。癌细胞会“改造”周围的正常细胞为自己服务——这正是“微环境重编程”概念的起点。

书中识别这些细胞类型靠的是 marker 基因（第 09 章正式使用），先混个脸熟：

```
# 全书统一的细胞类型 marker 基因（00-TECH 第 7 节）
markers <- list(
  肿瘤细胞 = c("EPCAM", "KRT8", "KRT19"),
  CAF      = c("COL1A1", "COL3A1", "DCN", "PDGFRA"),
  CD8_T    = c("CD3D", "CD8A", "GZMB"),
  巨噬细胞 = c("CD68", "CD163", "C1QA"),
  B 细胞   = c("MS4A1", "CD79A"),
  内皮细胞 = c("PECAM1", "VWF")
)
print(markers[["CAF"]])   # 查看 CAF 的 marker 基因
```

运行结果解读：输出"COL1A1" "COL3A1" "DCN" "PDGFRA"。这些基因在对应的细胞类型里高表达，是第 09 章注释 spot 的“字典”。

1.2 细胞通讯的基本形式：配体与受体

细胞之间的对话靠**配体-受体**（ligand-receptor, L-R）互作完成：一个细胞分泌或展示**配体**（ligand，信号分子），另一个细胞表面的**受体**（receptor）识别并结合配体，从而被“激活”。按配体如何到达受体，分为四种形式：

- **自分泌**（autocrine）：细胞分泌配体作用于自身，形成自我强化回路；
- **旁分泌**（paracrine）：配体扩散到邻近细胞，是肿瘤微环境中最常见的形式；
- **近分泌**（juxtacrine）：配体锚定在细胞膜上，必须与相邻细胞“面对面”接触才能结合；
- **内分泌**（endocrine）：配体经血液循环到达远处的靶细胞，在局部肿瘤微环境中较少见。

图 1-1 展示了最常讨论的三种模式。注意旁分泌与近分泌的区别在于配体是否“出门”：CXCL12 是分泌出去的，PD-L1 则是长在细胞膜上的。这个区别很重要——近分泌只有在两种细胞**空间相邻**时才可能发生，这正是本书强调空间分析的原因之一。

1.3 六条关键通讯通路

本书的教学通路全部来自 CellChat 数据库（CellChatDB.human，第 11 章详解），是真实存在的配体-受体对。先看看数据库里它们长什么样：

```
# 查看 CellChat 数据库里的教学通路（第 11 章正式使用）
if (requireNamespace("CellChat", quietly = TRUE)) {
```

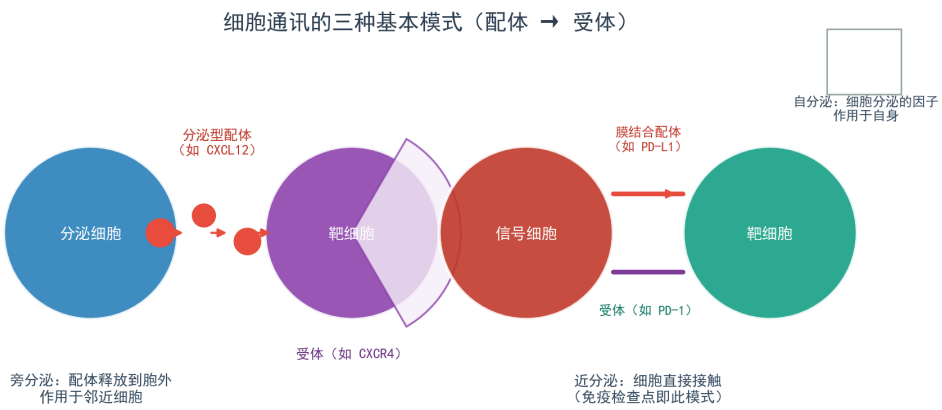

图 2: 图 1-1 细胞通讯的三种模式。旁分泌（paracrine）：分泌型配体作用于邻近细胞，如 CXCL12 → CXCR4；近分泌（juxtacrine）：膜结合配体与相邻细胞直接接触，如 PD-L1 → PD-1；自分泌（autocrine）：细胞分泌的配体作用于自身。

```
  db <- CellChatDB.human$interaction
  teach <- c("CXCL", "PD-L1", "TGFb", "VEGF", "MIF", "GALECTIN")
  print(db[db$pathway_name %in% teach,
           c("ligand", "receptor", "pathway_name")])
} else {
  message("CellChat 未安装，先看本节的表格即可。")
}
```

运行结果解读：输出六条通路的配体（ligand）、受体（receptor）与通路名（pathway_name）。你会发现数据库里的写法（如 TGFb）与正文写法（TGF-β）略有差异，这是命名习惯问题，不影响分析。

六条通路逐条解释（配体-受体对与 00-TECH 第 5 节一致）：

①趋化因子：CXCL12–CXCR4。 CXCL12（又称基质细胞衍生因子 SDF-1）主要由 CAF 与基质细胞分泌，受体 CXCR4 在癌细胞与部分免疫细胞表面表达。这条轴促进癌细胞迁移与“归巢”，是转移研究的热点，同时参与免疫细胞招募。

②免疫检查点：PD-L1（CD274）–PD-1（PDCD1）。肿瘤细胞表面高表达 PD-L1，T 细胞表面表达受体 PD-1。两者结合会“踩刹车”，抑制 T 细胞的杀伤功能——这是肿瘤免疫逃逸的核心机制，也是免疫治疗（抗 PD-1/PD-L1 抗体）的靶点。

③生长因子：TGF-β（TGFB1）–TGFBR2。 TGF-β 由肿瘤细胞与 CAF 分泌，通过受体 TGFBR2 传递信号，驱动**上皮-间质转化**（epithelial-mesenchymal transition, EMT）、免疫抑制与基质重塑，是侵袭前沿最活跃的通路之一。

④血管生成：VEGF（VEGFA）–VEGFR2（KDR）。肿瘤快速生长导致**缺氧**（hypoxia），缺氧诱导癌细胞分泌 VEGFA，激活内皮细胞上的受体 KDR（VEGFR2），刺激新生血管为肿

瘤供氧供营养。抗血管生成药物（如贝伐珠单抗）即靶向这条轴。

⑤炎症与免疫调节：MIF–CD74。巨噬细胞迁移抑制因子（MIF）与 CD74 结合，促进炎症、招募并极化巨噬细胞，与肿瘤进展和不良预后相关，是 CellChat 教学数据库中的代表通路之一。

⑥凝集素检查点：Galectin-9（LGALS9）–TIM-3（HAVCR2）。肿瘤与基质细胞表达 Galectin-9，与耗竭 T 细胞表面的 TIM-3 结合，诱导 T 细胞凋亡或功能耗竭，属于 “第二代” 免疫检查点，也是本书重编程分析的重点候选通路。

通路名	配体（ligand）	受体（receptor）	一句话含义
CXCL	CXCL12	CXCR4	趋化迁移、免疫招募
PD-L1	CD274	PDCD1	免疫检查点，抑制 T 细胞
TGFb	TGFB1	TGFBR2	EMT、免疫抑制、基质重塑
VEGF	VEGFA	KDR	血管生成
MIF	MIF	CD74	炎症与巨噬细胞调节
GALECTIN	LGALS9	HAVCR2	检查点，诱导 T 细胞耗竭

（说明：表头里的 CXCL、PD-L1、TGFb 等是 CellChatDB 的通路名，正文用带连字符的写法，两者指同一组互作。）

1.4 肿瘤如何 “重编程” 微环境

“重编程” 在这里指：肿瘤细胞通过持续发送信号，把正常组织改造成利于自己生长的 “土壤”。四条主线：

①免疫抑制。肿瘤通过上调 PD-L1、分泌 TGF-β 与 MIF 等，让杀伤性免疫细胞 “熄火”。例子：肿瘤细胞高表达 PD-L1，使浸润的 CD8 T 细胞进入功能抑制状态；免疫治疗通过解除这个刹车来恢复杀伤。

②血管生成。缺氧诱导 VEGFA 分泌，驱动内皮细胞增殖，肿瘤为自己搭建 “补给线”。例子：乳腺癌组织内可见大量新生血管，内皮细胞高表达 PECAM1、VWF（见 1.1 节 marker 表）。

③上皮-间质转化（EMT）。癌细胞 “脱下” 上皮特征、“穿上” 间质特征，获得迁移与侵袭能力。例子：侵袭前沿的癌细胞高表达 VIM、ZEB1，伴随 TGF-β 通路激活——这正是第 10 章识别 “侵袭前沿” 生态位的分子依据。

④促纤维化与基质重塑。肿瘤激活 CAF，后者大量分泌胶原等 ECM 成分，形成致密的 “物理屏障 + 免疫屏障”。例子：基质区 CAF 高表达 COL1A1、DCN，既是结构特征也是通讯枢纽。

这四条主线不是孤立的：EMT 与免疫抑制常在同一生态位（侵袭前沿）同时发生，形成 “恶性协作”——这正是本书 “通讯重编程” 要捕捉的系统性变化。

1.5 为什么“空间”对理解 TME 重要

图 1-2 是肿瘤微环境组成的理想化示意——注意“中央”与“周围”的差别：

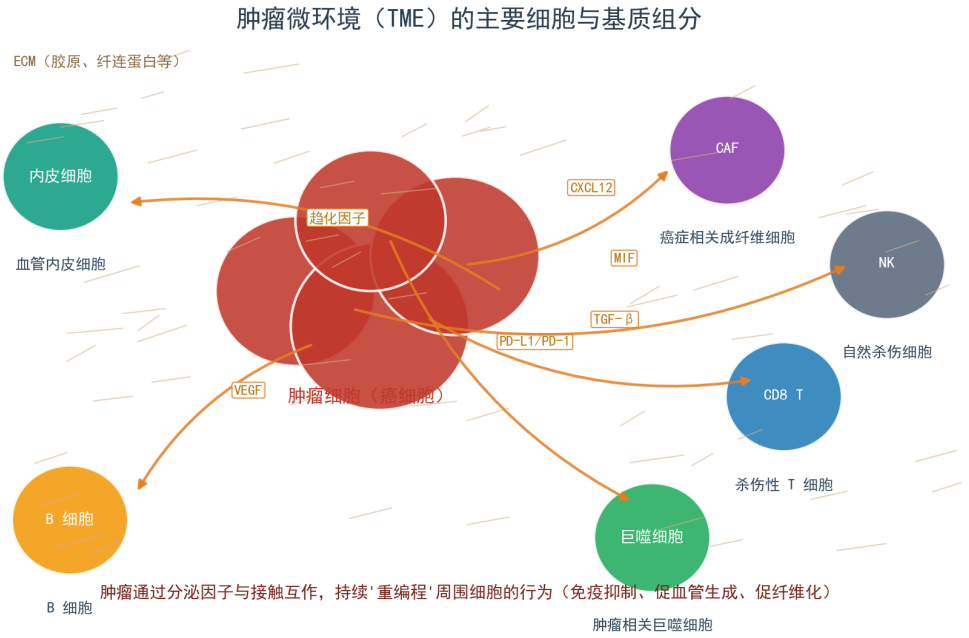

图 3: 图 1-2 肿瘤微环境组成示意。中央为肿瘤细胞团，周围分布 CAF、CD8 T 细胞、巨噬细胞、B 细胞、内皮细胞与 NK 细胞，背景为 ECM，箭头表示 CXCL12、TGF-β、PD-L1/PD-1、VEGF、MIF 等关键通讯。

真实组织里，不同区域的细胞组成差异巨大：肿瘤核心以癌细胞为主、缺氧、VEGF 高；侵袭前沿是肿瘤-基质交界、EMT 活跃；免疫浸润区聚集 T/B/巨噬细胞；基质区由 CAF 与 ECM 主导（特征对照见 00-TECH 第 4 节）。

如果只看全局平均，这些异质性会被彻底抹平。设想：某切片一半区域 PD-L1 高、一半区域几乎没有，全局平均会给出“中等水平”——既丢失了“哪里有”的信息，也丢失了“和谁共定位”的信息。而 PD-L1 是否与 CD8 T 细胞相邻（近分泌才可能发生），恰恰是免疫抑制是否真实起效的关键。

这正是本书选择“生态位视角”的根本原因：把空间切成有生物学含义的区域，再在区域内与区域间分析通讯。理解了这一点，你就抓住了整本书的灵魂。

本章小结

- 肿瘤微环境 = 癌细胞 + CAF + 免疫细胞 + 内皮细胞 + ECM，是一个动态生态系统；
- 通讯靠配体-受体互作，分自分泌、旁分泌、近分泌、内分泌四种形式；
- 六条教学通路（CXCL12–CXCR4、PD-L1–PD-1、TGF-β–TGFBR2、VEGF–KDR、MIF–CD74、LGALS9–HAVCR2）覆盖迁移、检查点、EMT、血管生成、炎症等核心过程；
- 肿瘤通过免疫抑制、血管生成、EMT、促纤维化四条主线“重编程”微环境；

- 空间位置决定通讯能否发生，全局平均掩盖异质性——生态位视角由此而来。

练习与思考

1.（动脑题）图 1-1 中，为什么 PD-L1–PD-1 属于 “近分泌” 而 CXCL12–CXCR4 属于 “旁分泌”？这对空间分析有什么启示？
2.（动脑题）TGF-β 同时出现在 “EMT” 与 “免疫抑制” 两条主线中，这提示不同生态位的通讯可能存在什么现象？
3.（动手题）运行 1.3 节的 R 代码；如果 CellChat 未安装，记录提示信息，然后手动把六条通路的配体-受体对写进一张表，与 00-TECH 第 5 节核对。
4.（思考题）如果某条通讯通路在数据中被预测为 “高活性”，你会如何设计实验去验证？（提示：参考 00-TECH 第 5 节的 “重要提醒”。）
5.（动脑题）为什么说 “通讯推断是计算预测，不是直接测量”？用 PD-L1–PD-1 举例说明。

第 02 章空间转录组技术原理：从单细胞到 10x Visium

本章回答三个问题：单细胞测序差在哪？空间转录组有哪些技术？本书用的 10x Visium 到底怎么工作？

本章目标

读完本章后，你应该能：

1. 说出单细胞测序的成就与“丢失空间位置”这个盲区；
2. 用表格对比原位捕获、原位成像、空间蛋白三条技术路线，并说出本书为什么聚焦 Visium；
3. 准确复述 10x Visium 的硬指标（捕获区、spot 数、直径、中心距）；
4. 说出 Visium 数据文件的四个组成部分及各自作用；
5. 说出空间转录组的三个局限，并解释“去卷积”为什么必要。

2.1 单细胞测序的成就与盲区

单细胞 RNA 测序（single-cell RNA sequencing, scRNA-seq）是过去十年生物学的里程碑：把组织解离成单个细胞，逐个测量转录组，从而回答“组织里有哪些细胞类型、各占多少、各表达什么”。它回答的问题是“**谁**在表达什么”。

但它有一个结构性的盲区：**解离把空间位置永久抹掉了**。解离后的细胞进了液滴或孔板，你无法知道它原来在组织的哪个位置——肿瘤核心的癌细胞和侵袭前沿的癌细胞，在解离后看起来“一样远”。而很多关键问题恰恰依赖位置：PD-L1 高表达的肿瘤细胞是否紧挨着 CD8 T 细胞？CAF 是否聚集在侵袭前沿？这些“空间关系”，单细胞数据回答不了。

空间转录组（spatial transcriptomics）的诞生正是为了补上这一课：不破坏组织，在切片上原位测量表达，每个测量点都带着坐标。

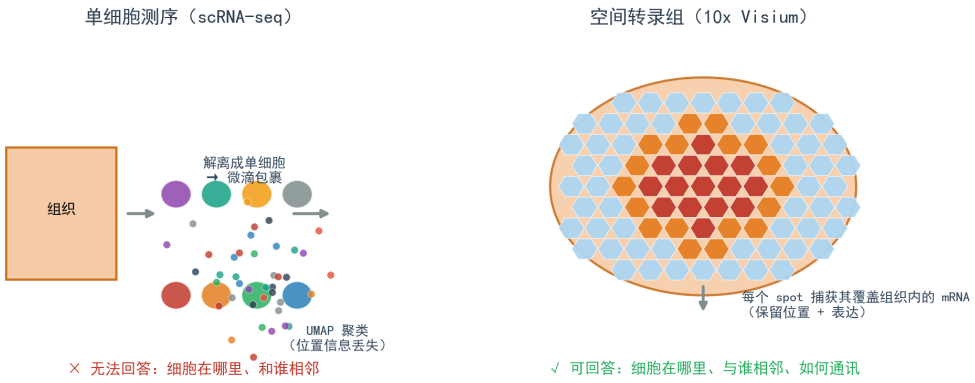

图 4: 图 2-1 单细胞测序与空间转录组的对比。左侧：组织被解离成单个细胞（液滴），聚类成 UMAP 后丢失了原始空间位置；右侧：组织切片保持原位，spot 网格上的每个点都保留坐标。

2.2 空间转录组技术谱系

空间转录组不是单一技术，而是一个家族，大致分三条路线：

- **原位捕获**（in situ capture）：用带空间条形码的捕获点“抓取”组织释放的 mRNA。代表：10x Visium（spot 网格）、Slide-seq（DNA 微珠，分辨率更高）、Visium HD（1.8 μm 六边形 bin）。
- **原位成像**（in situ imaging）：用荧光显微镜直接“看”组织里的转录本。代表：MERFISH、seqFISH、10x Xenium。分辨率可达单细胞甚至亚细胞，但一次只能测固定数量的基因（panel）。
- **空间蛋白**（spatial protein）：用抗体在组织上检测蛋白。代表：CODEX、MIBI。与转录组互补，但检测的蛋白数量有限。

技术路线	代表技术	分辨率	通量（基因数）	优点	缺点
原位捕获	10x Visium	spot 级（≈ 55 μm，含多个细胞）	全转录组	全转录组无偏、流程成熟、公开数据多	多细胞混合、分辨率有限
原位捕获	Slide-seq	微珠级（≈ 10 μm）	全转录组	分辨率更高	制备复杂、成本高
原位成像	MERFISH / seqFISH / Xenium	单细胞/亚细胞	数百至数千（panel）	分辨率高	基因数受限、需预设 panel
空间蛋白	CODEX / MIBI	单细胞级	数十至上百个蛋白	直接测蛋白	蛋白数有限、需抗体 panel

本书聚焦 10x Visium，原因有三：全转录组（无偏，不需要预设基因）、流程成熟（官方工

具链完善）、公开数据丰富（人乳腺癌 demo 等，第 07 章下载）。学会了 Visium，再看其他技术只是“换个尺子”。

2.3 10x Visium 原理详解

Visium 的核心思想一句话：**把“空间坐标”编码进测序读段**。原理分四步（对照图 2-2）：

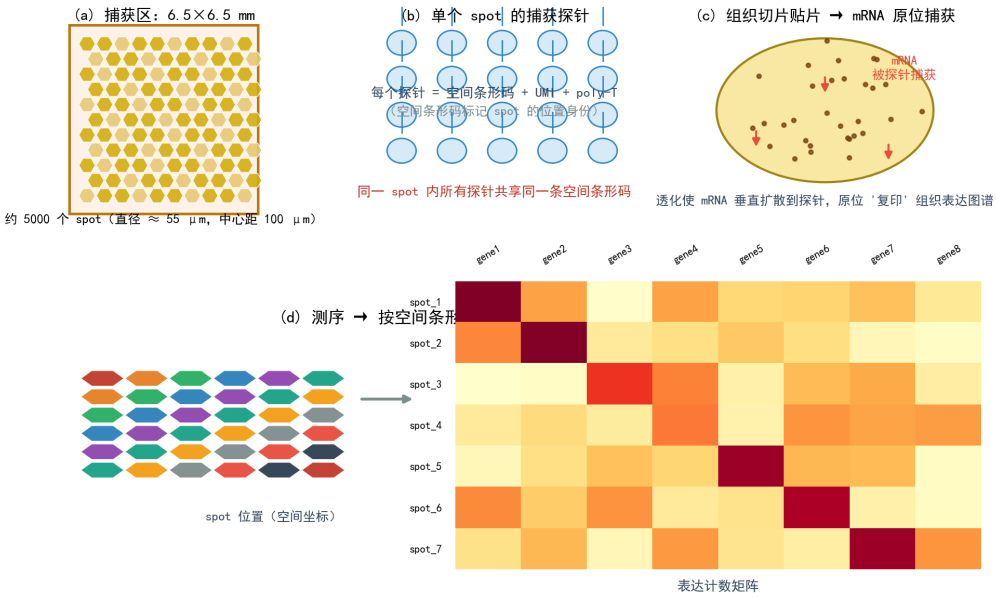

图 5: 图 2-2 10x Visium 原理四联图。(a) 捕获区 6.5 × 6.5 mm，约 5000 个六边形 spot（直径≈ 55 μ m、中心距 100 μ m）；(b) 单个 spot 上的捕获探针 = 空间条形码 + UMI + poly-T；(c) 组织贴片后 mRNA 原位捕获；(d) 测序后得到 spot × 基因表达矩阵。

1. **捕获芯片：**载玻片上有 4 个捕获区（capture area），每个 **6.5 × 6.5 mm**，内含约 **5000 个**六边形 spot。每个 spot 的**直径约 55 μ m，中心间距 100 μ m**。
2. **探针结构：**每个 spot 上密布捕获探针。探针 = 空间条形码（spatial barcode，16 bp，标记“我在哪个 spot”）+ UMI（唯一分子标识符，12 bp，标记“这是哪一条 mRNA 分子”）+ poly-T 尾巴。
3. **原位捕获：**把组织切片贴到芯片上、透化处理，mRNA 释放后被就近 spot 的 poly-T 捕获。**新鲜冷冻（FF）样本**直接捕获 mRNA 的 poly-A 尾巴；**FFPE 样本**则需先做探针杂交（Visium FFPE 流程，另需去石蜡与抗原修复），因为石蜡包埋会降解 mRNA、破坏 poly-A 尾。
4. **测序与矩阵：**洗去组织、原位逆转录、建库测序。每个读段读出“空间条形码 + UMI + 基因序列”，汇总后得到 **spot × 基因**的表达矩阵。

务必记住一个关键事实（第 08 章质控要用）：每个 spot 覆盖约 **1–10 个细胞**，所以 spot 本质是**多细胞混合信号**。Visium 的“分辨率”是 spot 级，不是单细胞级。

2.4 Visium 数据文件结构

从 10x 官网下载的 Visium 数据（第 07 章）包含两类内容：表达矩阵 + spatial 目录。

```
# 第 07 章下载并解压后，目录结构如下（示意；Windows 用 dir 查看）
data/visium/
├── filtered_feature_bc_matrix.h5   # 表达矩阵（h5 单文件版）
├── filtered_feature_bc_matrix/
│   ├── barcodes.tsv.gz             # spot 条形码（每行一个 spot ID）
│   ├── features.tsv.gz             # 基因（Ensembl ID、基因名等）
│   └── matrix.mtx.gz               # 稀疏计数矩阵（行 = 基因，列 =spot）
└── spatial/
    ├── tissue_positions_list.csv   # 每个 spot 的组织内坐标
    ├── scalefactors_json.json      # 图像缩放因子（坐标与像素换算）
    ├── tissue_hires_image.png      # 高分辨率组织 H&E 图像
    └── tissue_lowres_image.png     # 低分辨率组织图像
```

运行结果解读：这是第 07 章下载完成后的目录长这样（现在没有数据也没关系，先记住结构）。逐项说明：

- `barcodes.tsv.gz`：一列，每个 spot 的唯一 ID（形如 `AAACAAGTATCTCCCA-1`），就是矩阵的列名；
- `features.tsv.gz`：每行一个基因，含 Ensembl ID 与基因名。人类样本里有 `MT-` 开头的线粒体基因，第 08 章质控要用；
- `matrix.mtx.gz`：Market Exchange Format 的稀疏矩阵。绝大多数元素是 0（dropout，见 2.5 节），用稀疏格式存储以节省内存；
- `spatial/:tissue_positions_list.csv` 给出每个 spot 在组织切片上的像素坐标，`scalefactors_json.json` 用于坐标与不同分辨率图像之间的换算，两张 PNG 是组织 H&E 图像——空间可视化的“底图”。

加载数据通常用 h5 单文件版（第 08 章的 `Load10X_Spatial`），先预览一下维度：

```
# 如果已下载数据（第 07 章），可以预览矩阵维度
if (file.exists("data/visium/filtered_feature_bc_matrix.h5")) {
  library(Seurat)
  obj <- Load10X_Spatial(data.dir = "data/visium/",
                         filename = "filtered_feature_bc_matrix.h5")
  dim(obj)                  # 输出：基因数 × spot 数
} else {
  message("数据还没下载，先看文字说明（第 07 章会下载）。")
}
```

运行结果解读：人乳腺癌 demo 大约 36,000 个基因 × 约 5,000 个 spot。行数（基因）远大于

列数（spot），这是 Visium 数据的典型形态。

2.5 空间转录组的三个局限

局限一：spot 是多细胞混合（需要去卷积）。一个 spot 里可能有 1–10 个不同种类的细胞，测到的是它们的 “平均信号”。要回答 “这个 spot 里主要是肿瘤细胞还是免疫细胞”，需要**去卷积**（deconvolution）：借助单细胞参考数据，把混合信号拆回各细胞类型的比例。第 09 章的 cell2location 就是干这个的。

局限二：dropout 与低灵敏度。捕获效率有限，低表达基因经常测不到，矩阵里 0 非常多。这意味着 “没测到≠不表达”。分析时要用专门的统计方法处理稀疏性（第 08 章会讲），不能把 0 当绝对。

局限三：批次效应。不同切片、不同实验批次之间存在系统性差异。多切片比较（比如本书 “生态位之间” 的比较）之前必须做批次校正，否则差异可能来自技术而不是生物学。

记住这三个局限，你就理解了后面所有方法选择的动机：**去卷积解决分辨率、统计方法处理 dropout、批次校正保证可比性**。

本章小结

- 单细胞测序回答 “谁在表达什么”，但解离丢失了空间位置；
- 空间转录组三条路线：原位捕获（Visium、Slide-seq）、原位成像（MERFISH、seqFISH、Xenium）、空间蛋白（CODEX、MIBI），本书聚焦 Visium；
- Visium 硬指标：捕获区 6.5 × 6.5 mm、约 5000 个 spot、spot 直径≈ 55 μ m、中心距 100 μ m、每个 spot 含 1–10 个细胞；
- 数据 = barcodes + features + matrix（或 h5）+ spatial 目录（坐标、缩放因子、组织图像）；
- 三个局限：多细胞混合（去卷积）、dropout、批次效应。

练习与思考

1.（动脑题）为什么说 “spot 是多细胞混合信号”？这对第 09 章 “细胞类型注释” 提出了什么要求？
2.（动脑题）空间条形码（16 bp）与 UMI（12 bp）分别解决什么问题？芯片上约 5000 个 spot，16 bp 条形码（最多 2^16 = 65536 种）够用吗？
3.（动手题）去 10x Genomics 官网 Datasets 页面找到 Visium 人乳腺癌 demo，下载 “filtered feature-barcode matrix (H5)” 与 spatial 目录（详细步骤第 07 章），对照 2.4 节核对目

录结构。

4. （思考题）FFPE 样本为什么不能直接用 poly-dT 捕获？提示：石蜡包埋会降解 mRNA、破坏 poly-A 尾，探针法如何绕过这个问题？
5. （动脑题）如果换成 MERFISH 数据（数千基因的 panel），2.5 节的三个局限中哪些会变轻、哪些会变重？

第 03 章生信入门必备：R/Python 与核心数据结构

> 本章只讲 “够用 70%” 的基础。目标不是把你变成程序员，而是让你能读懂并修改本书的代码。

本章目标

读完本章后，你应该能：

1. 掌握 R 的向量、数据框、tibble、管道与函数写法；
2. 理解 ggplot2“数据 + 映射 + 几何层” 的绘图理念；
3. 掌握 Python 的 numpy 数组、pandas DataFrame 与列表推导；
4. 看懂 Seurat、SingleCellExperiment、AnnData 三个核心对象的结构；
5. 认识 10x mtx 三件套、h5、h5ad、rds、csv 等文件格式及对应读写函数；
6. 独立完成 20 分钟热身练习：读 csv →算均值→画第一张图。

3.1 R 语言快速上手

本书主流程用 R（Seurat、CellChat、BayesSpace），先补五个最小概念。每个概念只给 1–2 行示例，够用即可。

①向量（vector）：一维数据，用 `c()` 创建：

```
x <- c(3, 1, 4, 1, 5)
mean(x)          # 平均值 = 2.8
x > 2            # 逐元素比较，返回 TRUE/FALSE 向量
```

②数据框（data.frame）：表格数据，列是变量、行是观测：

```
df <- data.frame(spot = c("s1", "s2", "s3"),
                 nCount = c(3000, 5200, 1800))
df$nCount                # 用 $ 取列
```

```
df[df$nCount > 2000, ]   # 按条件筛选行
```

③ **tibble**：tidyverse 的“升级版”数据框，打印更友好、行为更一致，本书默认使用：

```
library(tidyverse)
tb <- tibble(spot = c("s1", "s2"), nCount = c(3000, 5200))
tb                          # 打印时显示类型，长表自动截断
```

④**管道 |>**：把左边的结果“喂给”右边的函数，让代码从“从内向外读”变成“从左向右读”，是本书的“主力语法”：

```
tibble(spot = c("s1","s2","s3"), nCount = c(3000, 5200, 1800)) |>
  filter(nCount > 2000) |>
  summarise(mean_nCount = mean(nCount))
```

⑤函数：名称 <- function(参数) { 函数体 }：

```
square <- function(v) v^2
square(c(1, 2, 3))   # 返回 1 4 9
```

运行结果解读：五段代码分别演示了取平均与逻辑比较、取列与筛选、tibble 打印、管道串联、自定义函数。第 08 章你会看到 `RunPCA(obj) %>% FindNeighbors(...)`——那是管道在真实分析中的延续（`%>%` 与 `|>` 功能等价，本书混用）。

3.2 R 绘图理念：ggplot2

ggplot2 的绘图哲学一句话：**图 = 数据 + 映射 + 几何层**。

- **数据（data）**：一个数据框，是图的“原料”；
- **映射（aes）**：把数据列“映射”到图形的视觉属性（x 轴、y 轴、颜色、大小）；
- **几何层（geom）**：决定画成什么形状（点、柱、箱线、密度……）。

```
library(ggplot2)
d <- data.frame(生态位 = rep(c(" 肿瘤核心", " 侵袭前沿"), each = 10),
                CXCL12 = c(rnorm(10, 5, 1), rnorm(10, 8, 1)))
ggplot(d, aes(x = 生态位, y = CXCL12)) +
  geom_boxplot() +
  labs(title = " 不同生态位的 CXCL12 表达")
```

运行结果解读： `aes(x=, y=)` 把两列映射到坐标轴，`geom_boxplot()` 画箱线图，`+` 号用来叠

加图层。本书后面所有图——气泡图、网络图、热图、空间散点——都是这个“数据 + 映射 + 几何”框架的变体，只是几何层不同。

3.3 Python 快速上手

本书的 Python 用于 scanpy、squidpy、cell2location（第 09、10、13 章）。三个最小概念：

① **numpy 数组**：同构多维数组，适合数值计算：

```
import numpy as np
a = np.array([[1, 2], [3, 4]])
a.mean()          # 全部元素均值 = 2.5
a.sum(axis=0)     # 按列求和 → array([4, 6])
```

② **pandas DataFrame**：带标签的表格，是 Python 界的“数据框”：

```
import pandas as pd
df = pd.DataFrame({"spot": ["s1", "s2", "s3"],
                   "nCount": [3000, 5200, 1800]})
df["nCount"].mean()          # 列均值
df[df["nCount"] > 2000]      # 按条件筛选行
```

③**列表推导**：一行生成列表的 Python 特色写法：

```
genes = ["CD3D", "CD8A", "GZMB"]
lower = [g.lower() for g in genes]   # 转小写
```

运行结果解读：numpy 用 `axis` 指定沿哪个轴计算，pandas 用方括号取列（`df["nCount"]`，对应 R 的 `df$nCount`），列表推导是“表达式 + for”的固定句式。记住“R 用 `$`、Python 用 `[]`”这个差异，跨语言读代码就不容易混。

何时用哪个？本书的策略是：主流程（预处理、注释、通讯）用 R，去卷积（cell2location）与空间统计（Squidpy）用 Python。两种语言都要能“读”，不必都精通。

3.4 两个生态的核心对象

真实分析中，你不会直接操作“裸”数据框，而是操作封装好的分析对象。R 侧是 **Seurat** 与 **SingleCellExperiment（SCE）**，Python 侧是 **AnnData**。

Seurat 对象的三个核心部件：

- **assay（测定）**：存表达矩阵，如 `RNA`（原始计数）或 `SCT`（SCTransform 归一化后）；
- **meta.data**：每个 spot/细胞的元数据表（样本、条件、聚类标签、QC 指标等）；
- **reductions**：降维结果（PCA、UMAP），供可视化与聚类使用。

AnnData 对象（scanpy 生态）的五个核心部件：

- **X**：表达矩阵（细胞 × 基因，与 Seurat 的 “基因 × 细胞” 互为转置）；
- **obs**：细胞元数据（对应 Seurat 的 meta.data）；
- **var**：基因元数据（Ensembl ID、高变基因标记等）；
- **obsm**：降维结果（X_umap、X_pca 等）；
- **uns**：非结构化信息（聚类配色、通路注释等）。

部件	Seurat	SingleCellExperiment	AnnData
表达矩阵	`obj[["RNA"]]`	`counts(sce)`	`adata.X`（注意转置）
细胞/spot 元数据	`obj@meta.data`	`colData(sce)`	`adata.obs`
基因元数据	`rownames(obj)`	`rowData(sce)`	`adata.var`
降维结果	`obj@reductions`	`reducedDim(sce)`	`adata.obsm`
自由信息	`obj@misc`	`metadata(sce)`	`adata.uns`

运行结果解读：三种对象结构高度对应（表达矩阵 + 细胞信息 + 基因信息 + 降维 + 杂项），只是命名不同。记住这张对照表，你在 R 与 Python 之间切换时心中有数。最容易踩的坑是维度方向：Seurat 是**基因 × 细胞**，AnnData 是**细胞 × 基因**。

3.5 常用文件格式

格式	是什么	典型后缀
10x mtx 三件套	稀疏表达矩阵 + 条形码 + 基因	`matrix.mtx.gz`、`barcodes.tsv.gz`、`features.tsv.gz`
h5（HDF5）	单文件打包的表达矩阵	`*.h5`
h5ad	AnnData 单文件	`*.h5ad`
rds	R 对象单文件	`*.rds`
csv/tsv	通用表格	`*.csv`

读取函数对照（R 生态用 Seurat，Python 生态用 scanpy）：

格式	R 读取	Python 读取
10x mtx 三件套	`Seurat::Read10X()`	`scanpy.read_10x_mtx()`
h5（HDF5）	`Seurat::Read10X_h5()`	`scanpy.read_10x_h5()`
h5ad	`zellkonverter`（间接）	`scanpy.read_h5ad()`
rds	`readRDS()`	不直接支持
csv/tsv	`read.csv()`、`readr::read_csv()`	`pandas.read_csv()`

运行结果解读：日常最常用的是三件套（Visium 默认下载格式）与 h5（单文件更省事）。10x Visium 下载的 h5 单文件名通常为 `filtered_feature_bc_matrix.h5`（见第 7 章）。h5ad 是 Python 生态的“存档格式”，rds 是 R 生态的——跨生态传递数据时，通常用 csv 或 h5 作为“通用语”。注意：mtx 是稀疏格式，千万不要用 Excel 打开；读进来之后你看到的才是完整矩阵。

运行结果解读：日常最常用的是三件套（Visium 默认下载格式）与 h5（单文件更省事）。h5ad 是 Python 生态的“存档格式”，rds 是 R 生态的——跨生态传递数据时，通常用 csv 或 h5 作为“通用语”。注意：mtx 是稀疏格式，千万不要用 Excel 打开；读进来之后你看到的才是完整矩阵。

3.6 20 分钟热身练习

把前面的概念串一遍：**造数据→读数据→算均值→画第一张图**。

```r
# 第 1 步：造一个小数据（20 个 spot 的 UMI 数，模拟）
set.seed(2024)
umi <- data.frame(
  spot   = paste0("spot_", 1:20),
  nCount = round(rnorm(20, mean = 3000, sd = 800))
)
write.csv(umi, "umi_demo.csv", row.names = FALSE)   # 存盘

# 第 2 步：读回来（真实分析中，这里读的是你的 Visium 数据）
umi <- read.csv("umi_demo.csv")

# 第 3 步：算均值
mean(umi$nCount)

# 第 4 步：画第一张图（直方图）
hist(umi$nCount, main = "UMI 数分布", xlab = "UMI 数", col = "skyblue")
```

运行结果解读：你会得到一个数值（约 3000，随机种子固定后每次运行结果相同）和一张直方

图。恭喜——你已经完成了“读数据→算统计→画图”的最小闭环，这正是第 08 章质控分析的雏形。

真实的“空间质控图”长什么样？提前剧透（模拟数据演示）：

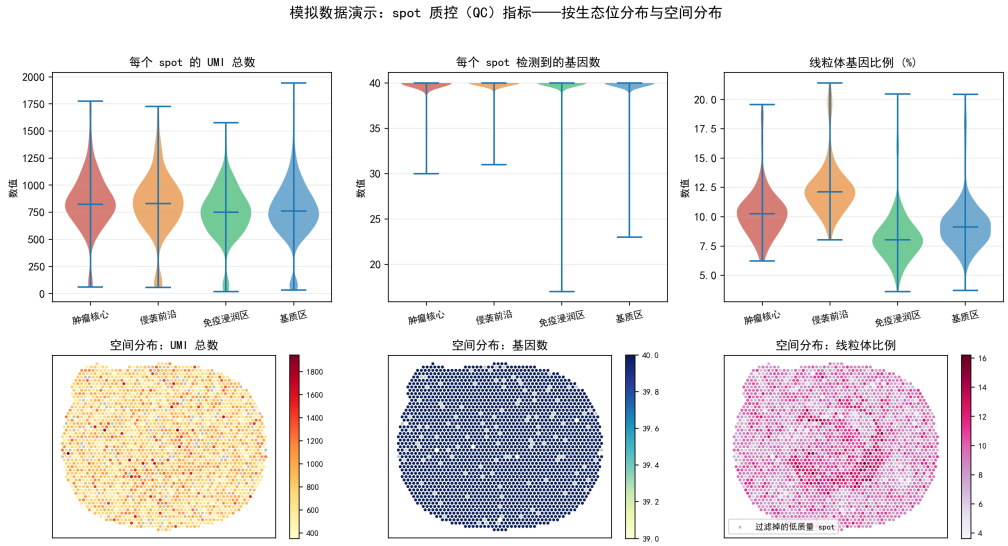

图 6: 图 3-1 空间质控图的“成品”样式（模拟数据演示）。上排：按生态位分组的 UMI 数、基因数、线粒体比例小提琴图；下排：三个指标映射回组织切片的空间散点，低质量 spot 标为灰色。这张图在第 08 章会详细讲解，现在你只需要读懂“每个点代表一个 spot”。

3.7 推荐学习资源

学有余力时，按顺序看这些（都是免费公开资源）：

- **R for Data Science（《R 数据科学》）**：tidyverse 圣经，官网 r4ds.hadley.nz，建议精读前 10 章；
- **《R 语言实战》**：中文经典教材，偏基础统计；
- **scanpy 官方教程**：scanpy-tutorials.readthedocs.io，跟着做 PBMC 案例即可入门；
- **Squidpy 文档**：squidpy.readthedocs.io，空间分析专属（第 10 章用到）；
- **10x Genomics 官网**：www.10xgenomics.com，Visium 原理与 demos 的第一手资料；
- 中文社区：生信技能树、生信宝典（搜索“Seurat 教程”“scanpy 教程”）。

原则：**遇到不懂的语法，先查官方文档，再搜社区**。本书第 16 章也整理了常见报错清单。

本章小结

- R 的核心：向量、数据框/tibble、管道 |>、函数；ggplot2 = 数据 + 映射 + 几何层；
- Python 的核心：numpy 数组、pandas DataFrame、列表推导；
- 三大对象：Seurat（assay、meta.data、reductions）、SCE（counts、colData、reducedDim）、AnnData（X、obs、var、obsm、uns），注意维度转置；

- 文件格式：三件套、h5、h5ad、rds、csv，各配读写函数；
- 20 分钟热身练习完成了 “读→算→画” 最小闭环，这是后续一切分析的原型。

练习与思考

1.（动手题）把 3.6 节练习的均值改为中位数（`median()`），并给直方图加上红色边框（`border = "red"`），观察变化。
2.（动手题）用管道把 3.1 节的 tibble 例子改成 “先筛选 nCount $>$ 2000 的 spot，再算均值”，并输出结果。
3.（动脑题）Seurat 的矩阵是 “基因 $\times$ 细胞”，AnnData 是 “细胞 $\times$ 基因”。如果跨生态读数据时不转置，会发生什么？
4.（动手题）用 Python 重写 3.6 节练习：pandas 读 csv、算均值、matplotlib 画直方图。
5.（思考题）为什么 h5ad 适合在 Python 生态 “存档”、rds 适合在 R 生态？跨生态传递数据时你更倾向哪种格式？

第 04 章空间生态位与通讯推断方法学总览

本章是全书方法学的 “地图”。从第 10 章到第 13 章的所有实操，都在这张地图上展开。你不需要现在就记住每个工具的参数，只需要建立三个东西：概念框架、方法谱系、选择逻辑。

本章目标

学完本章，你应该能够：

1. 准确区分**空间生态位（spatial niche）**、**空间域（spatial domain）**、**邻域（neighborhood）**三个概念，并说清它们的关系；
2. 说出生态位识别三大类方法（空间聚类、邻域分析、隐变量/多视图）各自由哪些代表工具构成、分别适合什么场景；
3. 区分 “无空间约束” 与 “空间感知” 两类细胞通讯推断方法，知道 CellChat、NicheNet、CellPhoneDB、COMMOT、Giotto 各是什么；
4. 用一张决策树为自己的课题选择方法组合；
5. 理解 “先分生态位、再做通讯” 为什么比全局通讯更有生物学意义。

4.1 三个核心概念：生态位、空间域与邻域

空间转录组文献里你几乎每天都会撞见 niche、domain、neighborhood 三个词，它们常被混用，但含义不同，先分清它们后面才不会糊涂。

空间生态位（spatial niche）。“生态位” 一词借自生态学：物种在生态系统中扮演的角色与位置。在空间转录组里，生态位指组织切片上由**特定细胞组成与微环境特征**反复出现的空间单位，强调**生物学功能**。本书的四个教学生态位——肿瘤核心、侵袭前沿（invasive front）、免疫浸润区、基质区——各有标志性细胞组合：肿瘤核心以肿瘤细胞为主，侵袭前沿是肿瘤细胞与成纤维细胞（CAF）交界、带 EMT 特征，免疫浸润区富集 T/B 细胞与巨噬细胞，基质区富集 CAF 与细胞外基质（ECM）。生态位是**解读性**概念：来自数据，但名字与含义需要人来赋予。

空间域（spatial domain）。更 “计算” 的词：指算法直接从数据划分出的、**转录组特征一致**

的连续空间区块，是算法的直接产物（BayesSpace 聚类得到的第 1 到第 q 个域）。很多文献把二者互换（本书 00-SPEC 亦注明），严格区分时：domain 强调“怎么算出来的”，niche 强调“是什么生物学结构”——domain 是 niche 的“计算候选”。

邻域（neighborhood）。与前两个词不在一个尺度上：指**以某个 spot（或细胞）为中心、一定半径内的 spot（或细胞）集合**，是微观、局部的概念，回答“谁经常挨着谁”——肿瘤细胞 spot 的邻居里，CAF 出现频率是否显著高于随机？邻域分析不直接分区，而是刻画**局部共定位模式**，为生态位组成提供证据。

概念	尺度	回答的问题	典型方法	强调
空间生态位（niche）	宏观区域	这块区域是什么生物学结构？	聚类 + 人工解读命名	生物学功能与组成
空间域（domain）	宏观区域	哪些 spot 转录组相似且空间连续？	BayesSpace、SpaGCN、STAGATE	计算划分
邻域（neighborhood）	微观局部	什么和什么经常挨在一起？	Squidpy nhood enrichment、co-occurrence	局部共定位

三者关系一句话概括：**邻域是微观证据，空间域是计算分区，生态位是有生物学解释的分区。**本书工作流：先空间聚类划出空间域，再邻域富集验证组成，最后结合先验知识把域命名为生态位——第 10 章会完整走一遍。

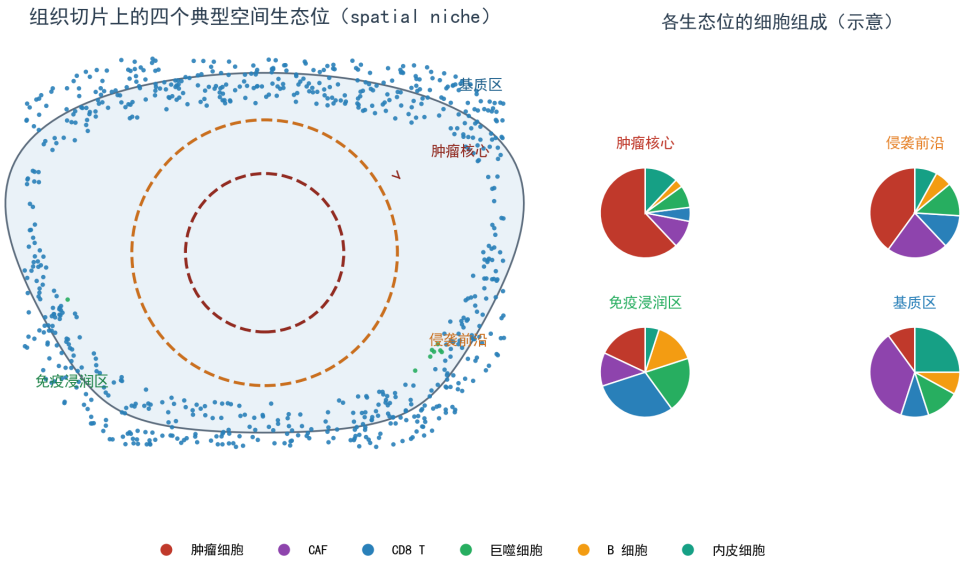

图 7: 图 4-1 空间生态位概念示意图：组织切片上四个生态位（肿瘤核心/侵袭前沿/免疫浸润区/基质区）的空间分布，右侧为各生态位的细胞组成饼图。

4.2 生态位识别方法谱系

生态位识别有三条技术路线，按“用什么信息、产出什么”区分。

4.2.1 空间聚类：把“相邻”写进模型

普通聚类（如 Leiden）只认表达矩阵，会把相邻但表达略异的 spot 分开、把不相邻但表达相似的 spot 合并；空间聚类则在模型里加入**空间先验**：相邻 spot 更可能属于同一区域。代表工具各一句话：

- **BayesSpace**（R，Bioconductor）：在聚类模型中加马尔可夫随机场式空间平滑先验，把“相邻 spot 倾向同类”编进贝叶斯框架，内置低分辨率增强。**适用场景**：要统计严谨的分区，想直接得到带空间先验的聚类标签。
- **SpaGCN**（Python）：图卷积网络联合**基因表达 + 空间坐标 + 组织学图像**学习，自动寻找“与组织学结构对齐的”域。**适用场景**：想利用 H&E 图像信息、组织结构清晰。
- **STAGATE**（Python）：图注意力自编码器，用空间自注意力学习 spot 间权重再聚类划分域。**适用场景**：通用首选之一，效果稳健、社区活跃。

三者输入相同（spot× 基因矩阵 + 坐标），差异主要在模型假设（贝叶斯先验/图卷积/注意力）与是否利用图像。先用 BayesSpace 预览这类方法的调用形态（完整实操见第 10 章）：

```r
# 预览（第 10 章详解）：BayesSpace 空间聚类最小流程
library(BayesSpace); library(SingleCellExperiment)
sce <- spatialPreprocess(sce, platform = "Visium", n.PCs = 15)
sce <- spatialCluster(sce, q = 7, platform = "Visium", d = 15)
clusterPlot(sce)
```

4.2.2 邻域分析：从“组成”验证“分区”

邻域分析不分区，而是统计**细胞类型之间的空间共定位**。Squidpy 的 `nhood_enrichment` 是代表作：对每一对细胞类型，检验“某类型的 spot 邻居中出现另一类型的次数”是否显著高于随机期望，输出 z 分数并做排列检验（permutation test），正 z = 共定位富集（红）、负 z = 互斥（蓝）。`co-occurrence` 则算共现分数随距离的衰减曲线，看两类细胞在多大距离内倾向共现。邻域分析是第 10 章生态位验证与全书空间证据的来源。

先用代码看一眼邻域富集与共现的调用方式（预览，第 10 章详解）：

```python
# 预览（第 10 章详解）：Squidpy 邻域富集与共现
import squidpy as sq
sq.gr.spatial_neighbors(adata, coord_type="generic", n_neighs=6)
sq.gr.nhood_enrichment(adata, cluster_key="celltype")  # 类型对共定位 z 分数
```

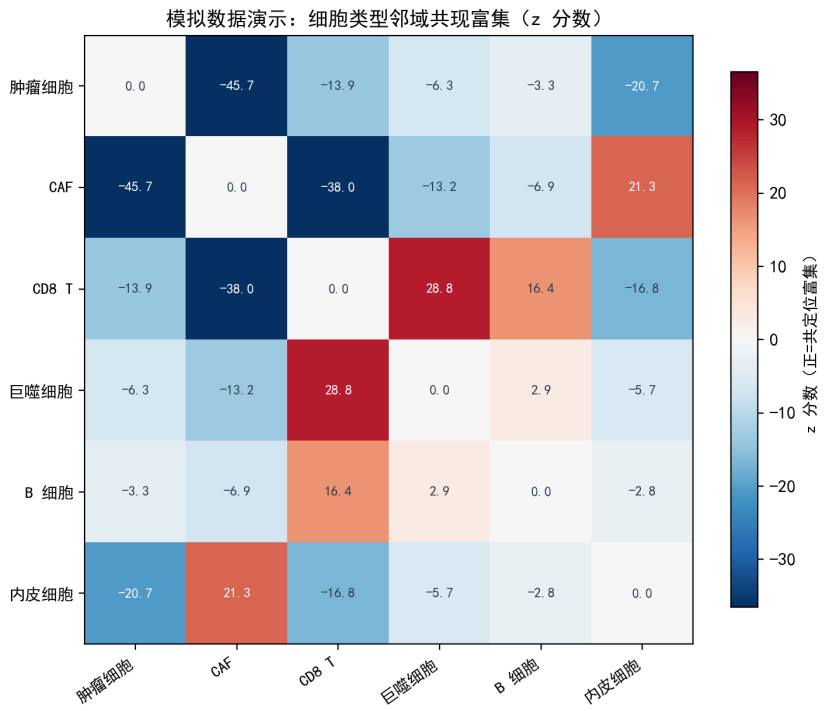

图 8: 图 4-2 邻域共现富集热图（Squidpy nhood_enrichment 风格，模拟数据演示）：细胞类型对之间的 z 分数，红色表示共定位富集，蓝色表示互斥，可据此判断生态位的典型细胞组合。

```
sq.pl.nhood_enrichment(adata, cluster_key="celltype")  # 富集热图（图 4-2 风格）
sq.gr.co_occurrence(adata, cluster_key="celltype")     # 共现分数随距离衰减
```

4.2.3 隐变量/多视图：解释“谁驱动了谁”

Misty（R 包 mistyR）不划区域，而是把每个基因的表达方差分解到**自身视图（intraview）**、**邻域视图（paraview）**、**旁区视图（juxtaview）**，用回归估计各视图解释的方差，回答“哪些基因的表达受空间环境驱动”。本书只介绍不实操（见 00-TECH 第 3 节），适合作为生态位发现后的“机制追问”工具。

方法	类型	语言	一句话原理	适用场景
BayesSpace	空间聚类	R	贝叶斯框架 + 空间平滑先验分区	统计严谨、需控制聚类不确定性
SpaGCN	空间聚类	Python	图卷积联合表达/坐标/H&E 图像	组织结构清晰、想用图像信息
STAGATE	空间聚类	Python	图注意力自编码器学空间权重再聚类	通用首选、效果稳健

方法	类型	语言	一句话原理	适用场景
Squidpy nhood_enrichment	邻域分析	Python	排列检验统计类型对共定位 z 分数	验证生态位组成、找共定位证据
Squidpy co-occurrence	邻域分析	Python	共现分数随距离衰减曲线	看共定位随距离变化
Misty	隐变量/多视图	R	多视图回归分解表达方差	追问空间环境对基因表达的贡献

4.3 细胞通讯推断方法谱系

细胞通讯（cell-cell communication, CCC）推断的基本思路：**配体-受体（ligand-receptor, L-R）数据库**（如 CellChatDB）给出 “谁和谁能对话”，表达矩阵给出 “双方在不在场”，据此计算通讯概率或活性分数。请记住：**这是计算预测，不是直接测量**——两个 spot 表达了配体和受体，不等于细胞真的在通信，须有共定位等空间证据佐证（见 4.4 与第 13 章）。

4.3.1 无空间约束：只看 “谁在表达什么”

这类方法只输入表达矩阵与细胞类型标签，**完全不用空间坐标**，可直接用于单细胞数据。

- **CellChat**（R）：基于质量作用定律（mass action）估计配体-受体互作的通讯概率，用排列检验评估显著性，再聚合到通路（pathway）层面。**下游分析完备**（网络、气泡图、差异比较），本书主线工具；v2 增加了空间模式（见下）。
- **NicheNet**（R）：走 “配体→受体→下游靶基因” 调控链，从**受配体影响的靶基因表达**反推哪些配体最活跃，回答 “哪个配体驱动了细胞状态变化” 而非仅 “谁在对话”。
- **CellPhoneDB**（Python）：统计检验配体-受体对（支持异源复合物）的表达共现，输出显著互作对，数据库由原研团队维护。轻量、直接，适合快速筛选候选互作。

以 CellChat 为例预览通讯分析主流程（第 11 章详解）：

```
# 预览（第 11 章详解）：CellChat v2 最小流程
cellchat <- createCellChat(object = obj, group.by = "celltype")
cellchat@DB <- CellChatDB.human
cellchat <- subsetData(cellchat)
cellchat <- identifyOverExpressedGenes(cellchat)
cellchat <- identifyOverExpressedInteractions(cellchat)
cellchat <- computeCommunProb(cellchat, type = "triMean")
cellchat <- computeCommunProbPathway(cellchat)
cellchat <- aggregateNet(cellchat)
```

4.3.2 空间感知：让“距离”参与计算

空间感知方法把坐标纳入模型，回答“哪些通讯在空间上真的可能发生”。

- **COMMOT**（Python）：用**最优传输（optimal transport）**把配体-受体信号在空间扩散，输出逐 spot 的**空间通信活性**（spatial communication score）与信号方向，直接给出“通讯发生在组织哪个位置”，是空间验证利器（第 13 章可选实操）。
- **CellChat v2 空间模式**（R）：createCellChat 识别含坐标的对象，提供 cell groups（按区域聚合）与 cell-spot（逐 spot）两种模式，把通讯信号**投影回组织切片**。本书第 11–13 章用它做生态位内通讯与差异比较。
- **Giotto**（R/Python）：一体化空间分析套件，内含空间通讯模块（表达加权 + 距离约束），适合想在一个框架内完成全流程的用户。

方法	语言	输入	输出	是否用空间信息	一句话定位
CellChat	R	表达矩阵 + 细胞类型	通讯概率/通路活性/网络	否（v2 可启用）	主流通讯分析全套工具
NicheNet	R	表达矩阵 + 靶基因	配体活性排序	否	从下游靶基因反推驱动配体
CellPhoneDB	Python	表达矩阵 + 细胞类型	显著 L-R 互作对	否	快速筛选候选互作
COMMOT	Python	表达矩阵 + 坐标 + L-R 库	逐 spot 空间通信活性	是	把通讯定位到组织位置
CellChat v2 空间模式	R	含坐标对象 + 细胞类型	空间通讯模式图	是	分区/逐 spot 通讯投影
Giotto	R/Python	表达矩阵 + 坐标	分区/通讯一体化结果	是	一体化空间分析平台

4.4 方法选择决策树：本书推荐组合

新手最容易犯的错是“全都要”。方法选择其实是一棵很短的决策树：

1. **第一步要什么？**要分区→空间聚类（BayesSpace/SpaGCN/STAGATE 三选一）；要先看组成→邻域富集。
2. **用哪个聚类？**想在 R 生态里顺手、要统计框架→ BayesSpace；想用深度学习→ STAGATE；想融合 H&E 图像→ SpaGCN。
3. **分区之后验证什么？**细胞共定位→ Squidpy nhood_enrichment。

4. **通讯怎么做？**生态位内网络与差异比较→ CellChat v2（分区模式）；想定位通讯发生位置→ COMMOT 空间活性。

本书推荐组合（也是第 10–13 章的执行顺序）：

环节	推荐方法	理由	对应章节
定生态位	BayesSpace + Squidpy 聚类/可视化	R 生态顺手、空间先验严谨	第 10 章
验证组成	Squidpy nhood_enrichment	提供共定位的统计证据	第 10 章
生态位内通讯	CellChat v2 分区模式	全套通讯分析 + 空间投影	第 11–12 章
差异/重编程	CellChat compareInteractions	生态位间系统比较	第 13 章
空间活性验证	COMMOT（可选）	把通讯定位到组织位置	第 13 章

伪代码把决策树翻译成逻辑（可运行代码见第 10–13 章）：

```
# 伪代码：方法选择决策树（仅示意逻辑，不可直接运行）
if 目标 == "划出空间区域":
    工具 = {"想融合 H&E 图像": "SpaGCN",
            "想要统计框架": "BayesSpace",
            "想用深度学习": "STAGATE"}
    标签 = 运行空间聚类（工具）
elif 目标 == "验证细胞组成":
    z 分数 = squidpy.nhood_enrichment(邻域图)
    热图（z 分数）   # 红 = 富集，蓝 = 互斥
elif 目标 == "生态位内通讯":
    通讯概率 = CellChat.computeCommunProb(按生态位分组)
    网络图（通讯概率）
elif 目标 == "空间活性验证":
    commot.tl.spatial_communication(坐标，距离阈值)
    空间散点（活性分数）
```

4.5 两张图读懂本书的生态位视角

图 4-1 展示“**生态位长什么样**”：切片被划分成四个生态位，每个右侧配细胞组成饼图。注意划分依据不只是“表达相似”，还包括**空间连续性**（相邻 spot 倾向同一生态位）与**生物学组成**（每个生态位的细胞组合可解读）——这就是生态位≠普通聚类的含义。

图 4-2 展示“**如何用证据支撑生态位**”：热图里细胞类型对的红蓝 z 分数显示哪些细胞倾向共聚，“肿瘤细胞 ×CAF”显著偏红即为“侵袭前沿由肿瘤与 CAF 交界构成”提供共定位证据。两张图合起来正是本书方法论闭环：**宏观分区（生态位）+ 微观证据（邻域富集）**。

4.6 为什么“先分生态位、再做通讯”更有生物学意义

你可能想问：直接在整个组织上跑一遍 CellChat 不就行了，为什么先分生态位？

因为**全局分析会把空间异质性“平均掉”**：肿瘤核心的肿瘤-肿瘤通讯、免疫浸润区的 T-B 互作、侵袭前沿的肿瘤-CAF 交界信号，在全局网络里混成一张“什么都有但什么都不突出”的大网，侵袭前沿特异的免疫抑制信号（PD-L1/PD-1、TGF-β）被肿瘤核心稀释，检验可能不显著。先按生态位分区、再在每个生态位内独立分析，相当于把镜头从“整座城市”拉近到“每个街区”——每个街区的通讯模式是自己的，差异（“重编程”）才可见。这正是本书核心卖点（communication reprogramming），第 12、13 章会反复回到这个逻辑。

本章小结

- 生态位（生物学分区）、空间域（计算分区）、邻域（局部共定位）是三个不同尺度的概念，全书主线是“邻域证据→空间域→生态位”。
- 生态位识别三路线：空间聚类（BayesSpace/SpaGCN/STAGATE）、邻域分析（nhood enrichment、co-occurrence）、隐变量/多视图（Misty）。
- 通讯推断两路线：无空间约束（CellChat、NicheNet、CellPhoneDB）与空间感知（COMMOT、CellChat v2 空间模式、Giotto）。
- 本书组合：BayesSpace/Squidpy 定生态位 + CellChat v2 分区通讯 + Squidpy 共定位验证 + COMMOT 空间活性。
- 通讯推断是计算预测，须配合空间共定位证据解读。

练习与思考

1. **概念题**：分别解释 spatial niche、spatial domain、neighborhood，各回答什么问题。
2. **判断题**：（a）nhood_enrichment 能直接输出生态位分区？（b）BayesSpace 属于空间感知通讯方法？（c）Misty 把表达方差分解到不同空间视图？
3. **选择题**：数据带清晰 H&E 图像且想利用图像信息分区，首选哪个工具？为什么？
4. **动手题**：安装 squidpy，用 `sq.datasets.visium_hne()` 加载示例数据，运行 `sq.gr.nhood_enrichment` 画富集热图，对照图 4-2 观察红蓝分布。
5. **讨论题**：为什么“两个 spot 分别表达配体与受体”不等于“细胞真的在通讯”？还能补充哪些证据？

第 05 章课题设计：创新点拆解与实验路线

很多新手拿到空间转录组数据后的第一反应是“赶紧跑 CellChat”。本章请你先按住手：花半天想清楚课题的一句话、三个创新点、三条可证伪的假设。设计想清楚了，后面十章的代码都是执行，而不是摸索。

本章目标

学完本章，你应该能够：

1. 用一句话模板写出自己的课题；
2. 说清“生态位视角、通讯重编程、空间共定位证据”三个创新层次分别新在哪里；
3. 把课题拆成可证伪的研究假设（H1/H2/H3），并知道什么结果会推翻它们；
4. 对照路线图说清每一步用哪个方法、对应本书哪一章、产出什么；
5. 避开课题设计的四个常见误区。

5.1 一句话课题模板

本书的课题可以压缩成一句话：

> **比较肿瘤组织不同空间生态位的细胞通讯网络，揭示微环境“通讯重编程”（communication reprogramming）。**

这句话可以当模板拆开看，四个成分各对应一种设计决策：

- **比较对象**：“不同空间生态位”——肿瘤核心、侵袭前沿、免疫浸润区、基质区（空间生态位的定义见第 4 章）。这决定了分析的单元是“生态位”而不是“整张切片”。
- **分析内容**：“细胞通讯网络”——配体-受体互作、通路活性、网络结构（第 11 章）。
- **核心结论**：“揭示通讯重编程”——生态位之间通讯的系统性差异，即“生态位特异（niche-specific）的通讯程序”（第 13 章）。
- **研究背景**：“肿瘤微环境”——生物学语境，决定了结果要往免疫抑制、EMT、血管生成等肿瘤学故事上落（第 1 章）。

把模板里的词替换成你自己的系统（比如结直肠癌、黑色素瘤），就得到你的课题一句话。注意：好的课题一句话应该**同时包含比较对象、分析内容、预期结论**，缺一个都容易被评审追问“所以呢？”。

5.2 创新点拆解：三个层次

5.2.1 第一层：生态位视角替代全局平均视角

常规分析把整个组织当成一个“大样本”，在全局水平比较肿瘤与正常、或比较不同病人。本书把镜头拉近：**以空间生态位为分析单元**。全局分析会平均掉空间异质性——侵袭前沿的免疫抑制信号会被肿瘤核心稀释（第 4 章 4.6 节已论证）。生态位视角的贡献在于让“位置”参与生物学结论：同一个信号，在肿瘤核心和侵袭前沿的含义可能完全不同。对应章节：第 10 章（生态位识别）。

5.2.2 第二层：通讯“重编程”而非单点通讯

比“谁在和谁说话”更进一步的问题是“**生态位之间的通讯程序有什么系统性差异**”。通讯重编程（communication reprogramming）在本教程中的操作化定义是：**生态位间通讯网络在数量、强度、通路层面的系统性差异**（第 13 章）。单点通讯（如“TGF-β 在肿瘤里上调”）随处可做；而“侵袭前沿整体切换到免疫抑制 + 侵袭相关的通讯程序，肿瘤核心则保持生长信号为主的程序”是**系统级、可对比、可讲故事**的结论，这是第二个创新点。

5.2.3 第三层：空间共定位证据增强

通讯推断本质是计算预测（第 4 章提醒过）。本书用**三重证据**给结论加固：①去卷积（deconvolution）确定每个 spot 的细胞组成（第 9 章）；②邻域富集（nhood enrichment）证明“配体细胞与受体细胞在空间上真的相邻”（第 10 章）；③空间通讯（CellChat v2 空间模式 / COMMOT 空间活性）把通讯信号定位回组织位置（第 11、13 章）。三重证据互相独立、互相印证，是审稿人最喜欢的“验证意识”，也是新手最容易被问倒的地方。

5.3 研究假设：可证伪的才是假设

课题设计的关键一步是把“我想看看有什么差异”升级为“我认为存在什么差异”——后者才可证伪。本书登记三条核心假设（与研究清单保持一致）：

编号	假设	可证伪判据（什么结果会推翻它）	对应章节/图
H1	不同空间生态位的通讯网络存在系统性差异（数量/强度/通路）	若各生态位通讯网络在数量、强度、通路三个维度都无显著差异（排列检验不显著）	第 12–13 章，fig13/fig14
H2	侵袭前沿生态位存在免疫抑制性通讯重编程（如 PD-L1/PD-1、TGF-β 高活性）	若侵袭前沿的免疫检查点与免疫抑制通路活性不高于其他生态位	第 13 章，fig14/fig15
H3	生态位分区通讯分析比全局分析揭示更多空间异质性信号	若全局分析能复现分区分析的全部显著互作、无新增信息	第 12 章，fig12/fig13

写假设时请注意三条纪律：①假设里**不出现“可能”“或许”**；②每一条都要写清“什么数据会让我认输”（上面第三列就是）；③假设个数 2–3 条为宜，贪多则验证成本失控。H2 里提到的 PD-L1（配体 CD274）/PD-1（受体 PDCD1）、TGF-β（TGFB1-TGFBR2）都是 CellChatDB.human 收录的教学示例通路（见 00-TECH 第 5 节），第 13 章会逐一检验。

5.4 分析路线图：从数据到论文

图 5-1 是全书路线图（fig01）的完整版。请把它贴在屏幕边上：每完成一章，就在图上划掉一段，你会很有成就感。

把路线图落到“步骤 × 方法 × 章节 × 产出”的表格，就是你的执行清单：

步骤	方法/工具	本书章节	预期产出
环境搭建	conda + R/Python 工具链	第 06 章	可复现环境（yml/lock）
数据获取	10x 官网 / spatialLIBD	第 07 章	原始数据 + 数据清单
预处理与质控	Seurat v5 / scanpy	第 08 章	质控图、过滤后的对象
细胞类型注释	marker 打分 + cell2location 去卷积	第 09 章	每 spot 细胞组成、类型空间图
生态位识别	BayesSpace + Squidpy nhood enrichment	第 10 章	生态位空间图 + 共定位富集热图

步骤	方法/工具	本书章节	预期产出
生态位内通讯	CellChat v2 分区模式	第 11–12 章	分生态位通讯网络与强度对比
通讯重编程	compareInteractions + COMMOT 空间活性	第 13 章	差异通讯气泡图、通路热图、Sankey
写作交付	图表规范 + 论文结构	第 14–15 章	论文级图表与结果叙述

把表格翻译成一段 “流水线骨架” 伪代码，贴在脚本最前面当目录用（不可直接运行，各步骤的可运行版本见对应章节）：

```
# 伪代码：本书课题的分析流水线骨架（对应图 5-1 路线图，不可直接运行）
# 各步骤的可运行版本见对应章节；伪代码仅用于看清 “每一步在做什么”

# ---- 第 06-07 章：环境与数据 ----
# conda activate spatial                  # 激活 Python 环境（第 6 章）
# source(”code/02_download.R”)            # 下载 Visium 数据（第 7 章）

# ---- 第 08 章：预处理与质控 ----
obj <- Load10X_Spatial(data.dir = ”data/visium/”,
                       filename = ”filtered_feature_bc_matrix.h5”)
obj <- subset(obj, subset = nFeature_Spatial > 200 &
               nCount_Spatial > 500 & percent.mt < 25)
obj <- SCTransform(obj, assay = ”Spatial”) |>
  RunPCA() |> FindNeighbors(dims = 1:30) |> FindClusters(resolution = 0.6)

# ---- 第 09 章：细胞类型注释（策略 A marker 打分 / 策略 B 去卷积）----
obj <- 注释细胞类型 (obj, markers = c(”EPCAM”, ”CD3D”, ”CD68”, ”COL1A1”))  # 伪代码

# ---- 第 10 章：生态位识别（BayesSpace 分区 + Squidpy 邻域富集验证）----
niche <- BayesSpace 聚类 (obj, q = 4)       # 伪代码：空间聚类划域
验证  <- nhood_enrichment(niche)          # 伪代码：共定位 z 分数验证组成

# ---- 第 11-12 章：生态位内通讯（CellChat v2 分区模式，逐生态位）----
net <- lapply(生态位列表, CellChat 分析)   # 伪代码：每个生态位独立建 CellChat

# ---- 第 13 章：通讯重编程（差异比较 + COMMOT 空间活性）----
diff   <- compareInteractions(前沿, 核心) # 伪代码：生态位间差异比较
commot <- spatial_communication(坐标)     # 伪代码：COMMOT 空间活性图

# ---- 第 14-15 章：论文级图表与结果写作 ----
# 输出 fig10-fig16 并组织成论文 Figure 2 版式（第 14 章）
```

图 9: 图 5-1 全书技术路线图：基础铺垫（01–04）→课题设计（05）→环境与数据（06–07）→预处理与注释（08–09）→生态位识别（10）→分区通讯（11–12）→通讯重编程（13）→写作交付（14–15）。

5.5 预期图表清单：先想好结果长什么样

课题设计阶段就把 “最终要交出哪些图” 列出来，能防止分析做到一半迷失方向。fig08–fig17 对应本书结果部分的 10 张核心图（均为模拟数据演示，真实分析长这样）：

图	一句话内容	出现章节
fig08_clustering	降维散点 + 聚类映射回切片的空间分布	第 08 章
fig09_celltype	每 spot 主要细胞类型的空间图 + 各生态位组成堆叠条形图	第 09 章
fig10_niche	生态位识别最终结果：切片按四种生态位着色	第 10 章
fig11_nhood	邻域共现富集热图：细胞类型对 z 分数	第 10 章
fig12_comm_network	各生态位并列的通讯网络圆图（边粗 = 强度）	第 11–12 章
fig13_comm_barplot	各生态位通讯数量与强度比较柱状图	第 12–13 章
fig14_diff_comm	生态位间差异通讯气泡图 + Δ 通讯概率热图	第 13 章
fig15_pathway_heatmap	生态位 × 信号通路活性热图	第 13 章
fig16_reprogramming	通讯重编程 Sankey 流量图（来源生态位→通路→靶细胞）	第 13 章
fig17_paper_figure	论文 Figure 2 版式：生态位 + 富集 + 差异 + 通路四联	第 14–15 章

把这张表当成你的 “图预算”：如果分析结束时某张图做不出来，说明对应的分析步骤还没走通；如果做出来但讲不出故事，说明设计阶段就没想清楚——现在补还来得及。

5.6 课题常见误区

- **误区一：贪多求全**。把 CellChat、NicheNet、CellPhoneDB、COMMOT、Giotto 全跑一遍，图做了一大堆，但每个都浅尝辄止。对策：围绕假设选工具（第 4 章决策树），一套主线（BayesSpace + CellChat v2 + COMMOT）走到底。
- **误区二：不设对照生态位**。只分析侵袭前沿，没有肿瘤核心或基质区做对照，“重编程” 就无从谈起——重编程是**生态位间的差异**，没有对照就没有差异。对策：至少选一个 “基

线” 生态位（通常肿瘤核心或基质区）做参照。

- **误区三：忽视空间验证**。表达层面看到 PD-L1/PD-1 上调就下结论，却没有邻域富集或空间共定位证据，审稿人一句 “这跟单细胞分析有什么区别” 就站不住。对策：把第 5.2.3 节的三重证据写进计划。
- **误区四：命名随意**。生态位名字（肿瘤核心、侵袭前沿……）必须由 marker 证据 + 组织学先验支撑，不能凭感觉贴标签（00-TECH 第 4 节也强调 “名称仅为教学约定”）。

5.7 练习：写出你的课题

动手题（建议 30 分钟完成，写作顺序别反）：

```
# 我的课题设计（模板）
## 课题一句话
比较 _ _ _ _ _（研究对象，如结直肠癌组织）不同 _ _ _ _ _（空间生态位）
的 _ _ _ _ _（细胞通讯网络/通路活性），揭示 _ _ _ _ _（通讯重编程/免疫抑制程序）。
## 假设 H1
_ _ _ _ _生态位的 _ _ _ _ _（通讯强度/通路）显著高于/低于 _ _ _ _ _生态位。
（判据：若 _ _ _ _ _则不成立）
## 假设 H2
_ _ _ _ _生态位存在 _ _ _ _ _（如免疫抑制性）通讯重编程，表现为 _ _ _ _ _
（如 PD-L1/PD-1、TGF-β 高活性）。
（判据：若 _ _ _ _ _则不成立）
```

本章小结

- 课题一句话模板 = 比较对象 + 分析内容 + 预期结论，四个成分缺一不可。
- 三个创新层次：生态位视角（分析单元）、通讯重编程（系统差异）、空间共定位证据（三重验证）。
- 三条可证伪假设 H1/H2/H3 各有明确的反证判据，对应第 12–13 章与 fig13–fig15。
- 路线图（fig01）把每一步对应到章节与产出，图预算（fig08–fig17）防止分析跑偏。
- 避开贪多求全、无对照生态位、缺空间验证、命名随意四个坑。

练习与思考

1. **动手题**：按 5.7 的模板完成 “我的课题一句话 + 两个可证伪假设”，判据一栏必须写具体（什么统计结果算推翻）。
2. **判断题**：以下哪句是可证伪假设？（a）“肿瘤微环境很复杂”；（b）“侵袭前沿的免疫抑制通路活性显著高于肿瘤核心”；（c）“不同生态位可能有些差异”。

3. **分析题**：为什么说“不设对照生态位就无法谈重编程”？请用 fig12 的三张并列网络图说明。
4. **选择题**：你的分析想证明“配体细胞与受体细胞在空间上相邻”，应该用哪个方法的哪个输出？
5. **讨论题**：如果 H3 被推翻（全局分析能复现分区分析的全部信号），你的课题还成立吗？该怎么调整叙述？

第 06 章环境搭建与工具链

本章解决 “代码跑在哪” 的问题。建议跟着本章把环境装好，再开始第 7 章的数据下载。装环境是新手最挫败的环节之一，但请放心：本章给出的命令在 Windows 和 macOS/Linux 上都可用（差异点单独标注），装完本章 6.6 节的验证代码能跑通，就说明环境合格。

本章目标

学完本章，你应该能够：

1. 解释为什么用 conda/mamba 管理 Python 环境，以及 renv 对 R 环境的作用；
2. 从零创建 Python 空间转录组环境（Python 3.10 + scanpy + squidpy + matplotlib）；
3. 安装 R ⩾ 4.3 生态：Seurat v5、CellChat v2、BayesSpace、tidyverse、patchwork、ComplexHeatmap；
4. 处理 Windows 用户特有的坑（Rtools、Bioconductor 依赖）；
5. 导出环境文件（conda env export、renv snapshot）实现可复现；
6. 用最小代码验证 Seurat、CellChat、squidpy 可用。

6.1 为什么用 conda/mamba 管理环境

空间转录组分析横跨 Python 与 R 两大生态，依赖几十上百个包，版本错一个就可能跑出 “灵异结果”：同样是 scanpy，1.9 和 1.10 的 API 不同；CellChat v1 与 v2 的用法差异更大（00-TECH 第 2 节特意标注了版本）。**可复现**的核心是 “环境即代码”：把环境内容导出成文件，任何人（包括三个月后的你自己）都能用一条命令恢复出一模一样的环境。

conda/mamba 的价值有三点：①隔离——每个项目一个环境，互不污染，装坏了直接删掉重建；②版本锁定——`conda env export` 把包名与版本号写进 yaml 文件；③跨平台——同一份 yaml 在 Windows 和 Linux 上都能恢复。mamba 是 conda 的 C++ 重写版，解决依赖的速度快得多，装生信包强烈建议用 mamba（安装方式见 00-TECH 第 2 节）。R 侧用 `renv` 做类似的 “项目级包仓库” 管理，见 6.5 节。

还有一个新手常问的问题：为什么不直接用系统 Python？因为系统 Python（尤其是 Windows 上）被大量软件共享，`pip install` 很容易装出版本冲突，升级系统包还可能破坏其他软件。

conda 的哲学是“一次项目、一个环境、随建随删”：实验做完环境可以删掉，需要时按 yaml 一分钟重建，完全不污染系统。这套哲学在生物信息领域已是事实标准，本书第 8 章之后所有 Python 操作都默认在 spatial 环境中进行。

6.2 创建 Python 空间转录组环境

打开终端（Windows 用 Anaconda Prompt 或 PowerShell），依次执行。代码里 `-c conda-forge` 指定通道（channel）很关键：conda 默认通道包不全、解析慢，而生信包的多数依赖在 conda-forge 与 bioconda 两个社区通道维护得最好。先装核心四件套（scanpy、squidpy、anndata、matplotlib），一次装太多包更容易触发依赖冲突。装包卡住时优先怀疑网络（代理）与通道顺序，见 6.7 表格。

```
# 用 mamba 创建独立环境（若没装 mamba，把 mamba 换成 conda 即可）
mamba create -n spatial python=3.10 -y

# 激活环境
conda activate spatial

# 安装空间转录组核心包（版本参考 00-TECH 第 2 节：scanpy>=1.10, squidpy>=1.4）
mamba install -c conda-forge scanpy squidpy anndata matplotlib pandas numpy -y

# 可选：cell2location（去卷积用，依赖 scvi-tools，训练建议内存 >=16 GB）
mamba install -c conda-forge -c bioconda cell2location -y

# 检查版本（应显示 scanpy 1.10+、squidpy 1.4+）
python -c "import scanpy; print('scanpy', scanpy.__version__)"
python -c "import squidpy; print('squidpy', squidpy.__version__)"
```

运行结果解读：最后一行应输出形如 `scanpy 1.10.x`、`squidpy 1.4.x` 的版本号，说明环境可用。若 `squidpy` 安装报依赖冲突，请把 `cell2location` 留到第 9 章前再装（它依赖较重，是可选组件）。cell2location 体积大、依赖多，新手可以暂时跳过，先用 marker 打分注释（第 9 章策略 A）。

补充一点：`python -c "import ..."` 这种“一行式验证”是判断环境可用与否的最快方法，第 16 章排错会反复用到。另外环境名可以自定义，但建议与项目绑定（本书统一用 spatial），并在所有脚本注释与笔记里保持一致，避免“我这个脚本到底在哪个环境跑过”的混乱。

6.3 安装 R 环境与关键包

安装 R 本身很简单：到 CRAN（The Comprehensive R Archive Network）下载 Windows 安装包（Linux 用系统包管理器），注意选择 **R ≥ 4.3** 并勾选 64 位；Windows 用户在安装 R

后必须先装 Rtools，见 6.4 节。RStudio 是社区推荐的免费集成开发环境，提供脚本编辑、控制台、变量查看与绘图面板，本书所有 R 操作默认在 RStudio 中完成。一个常见疑问是 “R 版本与包版本的关系”：Bioconductor 的包（如 BayesSpace）与 R 版本对齐，R 版本太老就装不上较新的 Bioconductor 包，所以本书把 R ⩾ 4.3 定为硬性要求。R 包安装代码（复制到 RStudio 控制台或脚本运行，单次约 10–30 分钟）：

```r
# ===== code/01_setup.R 核心部分：R 环境与包安装 =====
# 运行环境：R >= 4.3，建议在 RStudio 中逐段运行

# 1) 安装包管理器 BiocManager（Bioconductor 包的统一入口）
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

# 2) 基础包：tidyverse（数据处理）、patchwork（拼图）、devtools（装 GitHub 包）
install.packages(c("tidyverse", "patchwork", "devtools"))

# 3) Seurat v5（单细胞/空间分析主框架）
#    注意：v5 与 v4 语法有差异，见 00-TECH 第 2 节
install.packages("Seurat")

# 4) ComplexHeatmap（CellChat 的依赖之一，也从 Bioconductor 装）
BiocManager::install("ComplexHeatmap")

# 5) BayesSpace（空间聚类，第 10 章使用，Bioconductor 渠道）
BiocManager::install("BayesSpace")

# 6) CellChat v2（本书通讯分析主线，GitHub 开发版）
#    依赖 NMF 与 ComplexHeatmap，若提示缺失先装：
install.packages("NMF")
devtools::install_github("jinworks/CellChat")

# 7) 验证加载（全部应返回 " 载入需要的程辑包" 无报错）
library(Seurat)
library(CellChat)
library(BayesSpace)
library(patchwork)
```

运行结果解读：每行 `install.packages` 或 `BiocManager::install` 结束后若无红色 error 即成功；最后 `library()` 全部无报错说明安装完成。安装过程中出现的黄色 warning（警告）通常不影响使用，只有红色 error 需要处理；若 error 反复出现，把完整报错复制存档，方便后续检索。常见提示 “package ‘xxx’ is not available for this version of R” 通常意味着来源不对（普通包用 CRAN、Bioconductor 包必须走 BiocManager，见 6.7 表格）。

6.4 Windows 用户必看：Rtools 与依赖

Windows 上装 R 包有三个必踩的坑，提前处理可以省下几小时：

1. **Rtools 必须先装**。Windows 上编译 R 包需要工具链。请到 CRAN 的 “Rtools” 页面下载与你的 R 版本匹配的 Rtools（如 R 4.3 对应 Rtools43），安装时保持默认路径。**Bioconductor 包（BayesSpace 等）几乎必然需要 Rtools**，漏装会出现 “cannot find Rtools” 或编译报错。
2. **Bioconductor 包必须走 BiocManager**，不能 `install.packages("BayesSpace")`——普通 CRAN 渠道没有这些包。统一用 `BiocManager::install()` 即可自动处理依赖。
3. **CellChat 的依赖**：CellChat 依赖 `NMF`（非负矩阵分解）与 `ComplexHeatmap`，其中 `NMF` 依赖较多（如 `registry`、`rngtools`），`ComplexHeatmap` 必须从 Bioconductor 装。按 6.3 的顺序先装依赖再装 CellChat，可避免一半的报错。

安装 Rtools 后请确认 R 能找到它：在 R 里运行 `Sys.which("make")`，若返回空字符串，说明工具链没被识别，重启 RStudio 或手动把 Rtools 的 bin 目录加入系统 PATH 再试。注意 R 与 Rtools 的**大版本必须匹配**（R 4.3 配 Rtools43，R 4.4 配 Rtools44），混用会出现莫名其妙的编译错误。Windows 用户若报 “there is no package called ‘curl’ ” 之类的基础依赖缺失，先 `install.packages("curl")` 补上再装 Bioconductor 包。这条规则同样适用于 Linux 用户：部分系统库（如 libxml2、libcurl）缺失会导致 R 包编译失败，报错里出现 `configure: error` 时先用系统包管理器补齐再重试。

6.5 环境导出与复现

环境装好只是第一步，**导出环境文件才算完成**（conda-environments 技能强调 “锁定文件是复现基准”）：

```
# Python 侧：导出环境（在 spatial 环境激活状态下）
conda activate spatial
conda env export --no-builds > envs/spatial.yml        # 人类可读版
conda env export > envs/spatial.lock.yml               # 完整锁定版（含构建号）

# 恢复（换电脑/换人时执行，先建好 envs 目录）
conda env create -f envs/spatial.lock.yml
```

```
# R 侧：renv 项目级包管理（在项目根目录运行一次）
install.packages("renv")
renv::init()          # 初始化项目库，记录当前所有 R 包版本
# 分析中途新增包后，更新快照：
renv::snapshot()
# 恢复（新电脑上）：
```

```
renv::restore()
```

运行结果解读：`conda env export` 生成的两个 yml 文件与 `renv.lock` 都应提交进版本库（git）。之后任何人（包括你换电脑后）都能用 `conda env create -f` 和 `renv::restore()` 重建环境。注意 yml 文件里的包版本是安装当天的解析结果，**更新包后必须重新导出**，并顺手在 envs 目录写一行 CHANGELOG 记录改了什么。

还有一个常见问题：R 环境为什么推荐 renv 而不是也在 conda 里装 R？两种做法都可行，但 renv 更贴近 R 生态的习惯（Bioconductor 包、GitHub 包都能被记录），而 conda 里 R 包来源有限。本书约定：**Python 用 conda/mamba，R 用 renv**，各管各的依赖，互不干扰。环境文件（spatial.yml、spatial.lock.yml、renv.lock）是论文方法节 “软件与版本” 的素材——评审问 “分析环境怎么复现”，这三份文件就是答案。

6.6 验证安装：最小可运行测试

环境装完必须冒烟测试（smoke test）。下面三段代码分别验证 Python 侧、R 侧、CellChat 侧：

```
# 冒烟测试 1 (Python, 在 spatial 环境运行): squidpy 加载示例数据
import scanpy as sc
import squidpy as sq

adata = sq.datasets.visium_hne()   # 内置 Visium 示例数据（人淋巴结）
sq.gr.spatial_neighbors(adata, coord_type="generic", n_neighs=6)
print("squidpy OK, obs =", adata.n_obs, "genes =", adata.n_vars)
```

```
# 冒烟测试 2 (R): Seurat 读入内置数据并降维聚类
library(Seurat)
obj <- CreateSeuratObject(counts = pbmc_small[["RNA"]]$counts)
obj <- NormalizeData(obj) %>% FindVariableFeatures() %>% ScaleData() %>% RunPCA()
cat("Seurat OK, PCs =", ncol(Embeddings(obj, "pca")), "\n")
```

```
# 冒烟测试 3 (R): CellChat 能否加载数据库
library(CellChat)
db <- CellChatDB.human   # 加载人类配体-受体数据库
cat("CellChat OK, interactions =", nrow(db$interaction))
```

冒烟测试（smoke test）是 “装完即测” 的工程习惯：用最小代码确认核心功能可用，而不是等到第 8 章跑真实数据才发现环境问题。三段测试分别覆盖最容易出错的三个环节——Python 空间对象构建、Seurat 标准流程、CellChat 数据库加载，任何一段通过都说明对应依赖链完整。若

某段报错又不确定原因，把报错原文贴到搜索引擎或 R/Python 社区提问，附上 `sessionInfo()` 输出会大幅提高被解答的概率。

运行结果解读：三段分别输出 `squidpy OK`、`Seurat OK, PCs = 50`、`CellChat OK, interactions = ...` 即全部通过。任何一段报错，先对照 6.7 的坑表格排查，不要带着报错环境进入第 8 章。

6.7 常见安装坑与解法

现象	原因	解法
`cannot find Rtools`	Windows 未装 Rtools 或版本不匹配	到 CRAN 下载与 R 版本对应的 Rtools 并重装；R 4.3 → Rtools43
`package 'BayesSpace' is not available`	用了 CRAN 渠道	改用 `BiocManager::install("BayesSpace")`
CellChat 安装报缺 `NMF`/`ComplexHeatmap`	依赖未先装	先 `install.packages("NMF")` 与 `BiocManager::install("ComplexHeatmap")`，再装 CellChat
`devtools::install_github` 超时/失败	网络访问 GitHub 不稳	挂代理或重试；必要时用 `options(timeout = 600)` 调大下载超时
conda 解析依赖极慢/卡住	用了默认 channels	换 mamba；明确指定 `-c conda-forge`；或使用 `--solver libmamba`
Python 包装好后 `import scanpy` 报错	环境被污染或版本冲突	删环境重建：`conda env remove -n spatial` 后按 6.2 重来
`squidpy` 与 `cell2location` 依赖冲突	两包依赖矩阵重叠冲突	cell2location 单独建环境，或推迟到第 9 章再装
内存不足（Out of Memory）	Visium 单切片 Seurat 约需 2–6 GB（00-TECH 第 2 节）	关闭其他程序；cell2location 训练至少 16 GB 内存

表格按出现频率排序，前四行覆盖了 Windows 用户 80% 的安装报错。如果你遇到的问题不在表内，先判断 “报错是否与环境有关”（比如换一个包重装能否复现），再决定查环境还是查代码。

6.8 配套脚本说明

本章完整代码在配套脚本 `code/01_setup.R`（含分段注释与报错提示），正文 6.3 节给出的是其核心部分。Python 侧命令在 `envs/spatial.yml` 导出后即可复现。环境验证通过后，请做第 6.5 节的导出，并把 `spatial.yml`、`spatial.lock.yml`、`renv.lock` 一并交给版本库——这是你整个课题可复现的起点。

最后提醒两点关于“顺序”的纪律：其一，**环境先于数据**——第 7 章的数据有数百 MB，先装好环境再下载，避免下载完才发现环境装不上；其二，**报错先查环境**——后续章节任何“跑不通”，第一步都回到第 6.6 节的冒烟测试：先 `conda activate spatial`，再重跑版本检查命令。这能排除一半以上的疑难杂症（第 16 章排错清单也是这个顺序）。

图 10: 图 6-1 全书技术路线图（重复引用 fig01）：环境搭建位于“环境数据”阶段，是数据下载（第 7 章）与预处理（第 8 章）之前的前置步骤。

本章小结

- conda/mamba 提供 Python 环境的隔离、锁定与跨平台复现；renv 为 R 提供同样的能力。
- Python 侧：`mamba create -n spatial python=3.10` + scanpy $\geqslant$ 1.10 + squidpy $\geqslant$ 1.4 + matplotlib。
- R 侧：R $\geqslant$ 4.3 + RStudio，Seurat v5、CellChat v2（GitHub）、BayesSpace（Bioconductor）、

tidyverse、patchwork、ComplexHeatmap。

- Windows 三坑：Rtools、Bioconductor 走 BiocManager、CellChat 依赖 NMF/ComplexHeatmap。
- 环境导出（conda env export / renv snapshot）与冒烟测试是交付前的必做项。

练习与思考

1. **动手题**：完成 6.2 与 6.3 的环境安装，运行 6.6 的三段冒烟测试，把三个 OK 输出截图存档。
2. **动手题**：执行 6.5 的导出命令，把 `spatial.yml` 与 `renv.lock` 提交到你的版本库，并在 envs 目录写一行 CHANGELOG。
3. **判断题**：以下说法对错？（a）BayesSpace 可以用 `install.packages()` 安装；（b）R 4.2 的 Rtools 可以给 R 4.3 用；（c）conda env export 的 yml 可以直接恢复出逐字节一致的环境。
4. **思考题**：为什么 “环境即代码” 比 “我帮你装好了” 更可靠？请结合三个月后换电脑的场景说明。
5. **讨论题**：scanpy 1.9 与 1.10、CellChat v1 与 v2 的差异提示我们什么版本的坑？查阅本书 00-TECH 第 2 节，列出你需要注意的三处版本差异。

第 07 章数据获取：Visium 公开数据集

本章解决“分析什么数据”的问题。本书全部实操基于 10x Genomics 官网公开的人乳腺癌 Visium demo 数据（组织结构典型、生态位分明、可免费下载）。数据下载看似简单，但“从哪来、多大、校验值多少、能不能用”这四个问题必须留下书面记录——这是 data-inventory（数据清单）的基本纪律，也是你论文方法节要写的素材。

本章目标

学完本章，你应该能够：

1. 说明为什么选 10x Visium 人乳腺癌 demo 作为主示例数据，以及结直肠癌数据何时作为拓展；
2. 走通 10x 官网的下载流程，说清每个下载文件的用途；
3. 认识三条备选数据渠道：spatialLIBD、GEO、Zenodo，知道各自适合什么场景；
4. 用 data-inventory 思想为数据建立清单（来源、URL、大小、md5 校验值）；
5. 按 data/raw 与 data/processed 组织目录，运行配套下载脚本 code/02_download.R。

7.1 数据集选型：为什么是人乳腺癌 demo

选示例数据要同时满足四个条件：**公开可下载、组织结构典型、生态位分明、体积适中**。10x Visium 人乳腺癌（Human Breast Cancer）demo 完美满足：

- **组织结构典型**：乳腺癌组织切片上肿瘤核心（tumor core）、基质区（stroma）、免疫浸润区（immune-infiltrated region）在组织学上清晰可辨，还常有侵袭前沿（invasive front）——正是本书四个教学生态位（第 4 章）的自然载体。
- **生态位分明**：乳腺肿瘤异质性强，导管原位癌/浸润性癌区域的肿瘤细胞密度差异大，免疫浸润与基质成分的空间分布分明，做“生态位 × 通讯”分析信号足、故事好讲。
- **社区成熟**：该数据集是空间转录组方法学的“标准考题”，大量工具论文用它做演示，遇到问题容易搜到答案。
- **体积友好**：单个 Visium 切片约 5000 个 spot、约 2 万个基因（见 00-TECH 第 1 节），H5 文件数百 MB 量级，个人电脑可流畅分析。

从生物学角度看，乳腺癌也是研究空间异质性的理想模型：浸润性癌区域肿瘤细胞密集、增殖信号强；导管原位癌保留导管结构；间质区域富含成纤维细胞与细胞外基质；免疫浸润区域（肿瘤浸润淋巴细胞）的存在与预后及免疫治疗响应密切相关。这些区域在空间上交错分布，意味着同一张切片上就能同时观察到 “生长、侵袭、免疫、基质” 四种微环境程序，天然适合 “生态位 × 通讯” 的分析框架；结构相对均一的组织（如某些正常器官）做生态位分析的信号会弱很多。

作为拓展，**结直肠癌**（10x 官网亦有 Visium demo）组织结构同样典型且免疫浸润丰富，适合想验证方法普适性的读者——把第 5 章课题模板里的 “肿瘤组织” 换成 “结直肠癌组织” 即可。黑色素瘤、人淋巴结等也可按需替换（00-TECH 第 6 节）。

7.2 10x 官网下载流程

下载分六步，全程不需要登录：

1. **打开 10x Genomics 官网**，进入 “Datasets” 页面（Products 下拉菜单→ Datasets，或直接搜索 “10x Genomics Datasets”）。
2. **筛选 Visium 数据**：在页面左侧筛选器中选择产品线 **Spatial → Visium**（如果要做 FFPE 数据，可选 Visium FFPE；本书以新鲜冷冻 Visium 为例）。
3. **找到 Human Breast Cancer**：在列表中定位 “Human Breast Cancer, Ductal Carcinoma In Situ, Invasive Carcinoma (FFPE/FF 均有，选 Fresh Frozen 即可)”，点击进入详情页。列表里每个数据集都标注了组织类型、物种与文件清单。
4. **下载表达矩阵**：点击 **filtered feature-barcode matrix (H5)** 下载单个 H5 文件。这个文件同时包含三个信息：spot 条形码（barcodes）、基因（features）、计数矩阵（matrix），是 `Load10X_Spatial()` 的入口（第 8 章）。
5. **下载空间信息**：下载 **spatial** 目录（官网提供为压缩包，常见 tar.gz 或 zip，解压后内含 4 个文件，见下表）。这一步最容易漏——没有它，数据就没有坐标和图像，无法做任何空间分析。
6. **解压并登记**：把下载内容解压到 data/raw/ 下，并立即按 7.4 节登记清单（含 md5）。

下载前提醒三件事：①官网界面会改版，筛选器名称可能有出入，认准 “Visium” “Human Breast Cancer” “filtered feature-barcode matrix (H5)” 三个关键词即可；②大文件建议在稳定网络下下载，浏览器中断可改用 7.6 的 curl 命令；③两个文件要放在同一目录且保持文件名不变，第 8 章 `Load10X_Spatial()` 会按文件名自动识别。

各文件用途如下（与 00-TECH 第 1 节一致）：

文件	内容与用途
`filtered_feature_bc_matrix.h5`（根目录）	spot× 基因计数矩阵（已过滤），Seurat/ scanpy 读入表达数据

文件	内容与用途
`spatial/tissue_positions_list.csv`	每个 spot 的像素坐标与组织内/外标记，空间可视化与邻域计算
`spatial/scalefactors_json.json`	图像与坐标之间的缩放因子，把表达矩阵对齐到组织图像
`spatial/tissue_hires_image.png`	高分辨率 H&E 组织图像，高分辨率叠加可视化
`spatial/tissue_lowres_image.png`	低分辨率 H&E 组织图像，快速预览与默认叠加背景

7.3 备选数据源

官网下载不是唯一途径。按 00-TECH 第 6 节，还有三条常用渠道：

- **spatialLIBD**（Bioconductor 包）：专用于人脑背外侧前额叶（DLPFC）Visium 数据（Maynard 等 2021 发表的数据集），一条命令即可拉取，适合想在 “金标准手工注释” 数据上练习生态位识别的读者。示例：`spatialLIBD::fetch_data("visium_IFDLPFC")`（首次运行会提示下载目标目录）。注意它只有脑组织数据，生态位类型与肿瘤不同。
- **GEO**（Gene Expression Omnibus）:NCBI 的公共表达数据库,用关键词 “Visium” “spatial transcriptomics” 检索即可找到大量已发表的空间转录组数据集（如乳腺癌、结直肠癌队列）。GEO 下载是文本化的三件套（barcodes/features/matrix）或补充文件，需要自己整理目录结构——适合想用 “真实队列” 重跑本书流程的进阶读者。下载时务必记录 GEO accession 号与下载日期。
- **Zenodo**：通用科研数据仓库，很多空间组学论文把补充数据（含处理后的 AnnData 对象）放在这里。检索时用数据集标题或 DOI 定位，**以论文方法节或数据可用性声明（Data Availability）给出的链接为准**，不要凭记忆猜 accession 号。

选渠道的决策很简单：**教学练习用官网 demo，方法对比用 spatialLIBD，真实课题用 GEO/Zenodo 的队列数据**。无论哪个渠道，都要执行 7.4 的登记纪律。

补充一个判断标准：**先看论文的数据可用性声明（Data Availability），再看文件格式**。声明写 “GEO accession GSE…”，就去 GEO；写 “processed data on Zenodo”，就去 Zenodo；写 “available via spatialLIBD”，就一条命令拉取。文件格式上，h5/h5ad 最省事，mtx 三件套需要自己整理，原始测序数据（fastq）则要重跑上游流程——除非你想做对比分析，否则直接下载处理后的表达矩阵即可。

7.4 数据清单与校验（data-inventory）

数据下载后第一件事不是跑分析，而是登记。data-inventory 的核心思想是：**每份数据一条记录，含来源、下载命令、校验值、大小、授权**，写入 `data/README.md`（或 manifest.tsv），保证数据丢失时可恢复、论文可追溯。本书示例登记如下（URL 以你在官网实际复制到的为准）：

字段	内容（示例）
id	visium-breast-dcis-invasive
类型	raw / public
来源	10x Genomics Datasets 官网（Visium → Human Breast Cancer）
下载命令	`curl -L -O <官网直链>`（见 7.6 脚本）
文件	filtered_feature_bc_matrix.h5、spatial.tar.gz
大小	约数百 MB（H5 为主）
校验值	md5: 以 `Get-FileHash` / `md5sum` 实际计算为准（见下）
授权	10x 公开 demo 数据集，公开可用，论文注明来源与下载日期即可
状态	active

登记表建议用 `data/README.md`（人读友好）或 `data/manifest.tsv`（机器友好）二选一，全书统一即可。另外建一张 `data/samples.csv`：一行一个样本，列为 sample_id、来源、下载日期、处理状态——它是后续所有分析脚本引用样本的“唯一事实来源”，第 8 章起脚本都从这里读样本信息，而不是硬编码路径。

校验命令（Windows PowerShell 与 Linux/macOS 各一条，二选一）：

```
# Linux/macOS: 计算 md5
md5sum data/raw/visium-breast/filtered_feature_bc_matrix.h5

# Windows PowerShell: 计算 md5 (输出即登记表里的校验值)
Get-FileHash -Algorithm MD5 data/raw/visium-breast/filtered_feature_bc_matrix.h5
```

运行结果解读：输出一串 32 位十六进制字符串。把它写进数据清单的“校验值”列；下次数据异常时重算并比对，就能判断是不是文件损坏。**先记校验值、后跑分析**，是数据管理的黄金顺序。

7.5 目录组织建议

本书统一采用以下目录结构（第 8 章起所有脚本都按这个约定读写）：

```
spatial-niche-tutorial/
├── data/
```

```
│   ├── raw/                        # 原始下载，只读，绝不修改
│   │   └── visium-breast/
│   │       ├── filtered_feature_bc_matrix.h5
│   │       └── spatial/            # 坐标、缩放因子、H&E 图像
│   ├── processed/                  # 预处理产物（QC 后对象等，第 8 章起生成）
│   ├── README.md                   # 数据清单（7.4 节登记表）
│   └── samples.csv                 # 样本清单（样本 id、来源、下载日期）
├── code/                           # 分析脚本
├── figures/                        # 输出图
└── envs/                           # 环境文件（第 6 章）
```

纪律只有两条：**data/ 不进版本库**（大文件 + 原始数据，git 只留清单与样本表）；**raw/ 只读**——任何修改都在 processed/ 里做，保证原始数据可随时重来。

为什么坚持 “raw 只读”？因为原始数据是分析的唯一锚点：第 8 章的过滤阈值、归一化方式改来改去很正常，但只要 raw/ 不动，任何时候都能从原始矩阵重新开始，所有中间结果都可复现。若某次处理真的需要覆盖，先备份到 processed/ 再改，并在 README 留一行记录（改了什么、何时改的、为什么）。

7.6 下载脚本：code/02_download.R 核心部分

完整脚本见配套 `code/02_download.R`（含断点续传与错误处理），核心部分如下。脚本会自动建立目录、下载、解压、计算校验值并追加到 `data/README.md`：

```r
# ===== code/02_download.R 核心部分 =====
# 用法：把 URL 换成你在 10x 官网复制到的直链（页面上的 Download 按钮右键复制链接）
# 注意：H5 与 spatial 的直链会随官网改版变化，以下 URL 为占位示例
url_h5    <- paste0("https://cf.10xgenomics.com/samples/spatial-exp/",
                    "filtered_feature_bc_matrix.h5")  # URL 以官网实际为准
url_sp    <- "https://cf.10xgenomics.com/samples/spatial-exp/.../spatial.tar.gz"
dest_dir  <- "data/raw/visium-breast"

dir.create(dest_dir, recursive = TRUE, showWarnings = FALSE)

# 1) 下载表达矩阵（H5）与空间目录
download.file(url_h5,
              file.path(dest_dir, "filtered_feature_bc_matrix.h5"),
              mode = "wb")
download.file(url_sp, file.path(dest_dir, "spatial.tar.gz"), mode = "wb")

# 2) 解压 spatial 目录
untar(file.path(dest_dir, "spatial.tar.gz"), exdir = dest_dir)
```

```r
# 3) 计算 md5 校验值 (Windows 与 Linux 通用写法)
md5_h5 <- tools::md5sum(file.path(dest_dir, "filtered_feature_bc_matrix.h5"))
cat("H5 md5:", md5_h5, "\n")   # 把这串值记进 data/README.md

# 4) 快速检查文件是否齐备
stopifnot(file.exists(file.path(dest_dir, "filtered_feature_bc_matrix.h5")),
          file.exists(file.path(dest_dir, "spatial/tissue_positions_list.csv")))
cat(" 下载与解压完成，文件齐备。\n")
```

运行结果解读：脚本最后打印 md5 值与“文件齐备”，说明下载成功。如果下载速度慢或中断，可改用浏览器手动下载后放到同一目录（脚本第 3、4 步仍可复用）。

脚本里的 URL 是占位符，务必替换成你在官网复制的直链——10x 的下载链接包含样本专属路径，不同数据集、不同版本都不一样。若官网改版导致直链失效，最稳妥的办法是回浏览器手动下载，再放到脚本指定的 data/raw/visium-breast/ 目录，脚本第 3、4 步（校验与检查）依然可用。下载完成后顺手把下载日期记进 data/README.md（data-inventory 要求记录下载日期）。命令行直接下载的替代写法（bash）：

```bash
# 等价于脚本第 1 步 (bash 版，URL 同样换成官网直链)
mkdir -p data/raw/visium-breast
curl -L -o data/raw/visium-breast/filtered_feature_bc_matrix.h5 "<H5 直链 >"
curl -L -o data/raw/visium-breast/spatial.tar.gz "<spatial 直链 >"
```

7.7 练习：下载并登记

1. 按 7.2 的六步在 10x 官网下载乳腺癌 demo 的 H5 与 spatial 目录；
2. 用 7.6 的脚本（或手工）解压到 `data/raw/visium-breast/`；
3. 用 7.4 的校验命令计算两个文件的 md5，填进 `data/README.md` 的登记表；
4. 回答：如果三个月后 H5 文件损坏，你靠哪条记录能恢复数据？

本章小结

- 主示例数据：10x Visium 人乳腺癌 demo（组织结构典型、生态位分明、公开免费）；拓展可选结直肠癌。
- 下载六步：Datasets 页面→ Visium 筛选→ Human Breast Cancer → H5 表达矩阵→ spatial 目录→解压登记。
- 备选渠道：spatialLIBD（人脑，一条命令）、GEO（真实队列）、Zenodo（论文补充数据），按用途选择。
- data-inventory 纪律：来源/URL/大小/md5/授权逐项登记进 data/README.md，先记

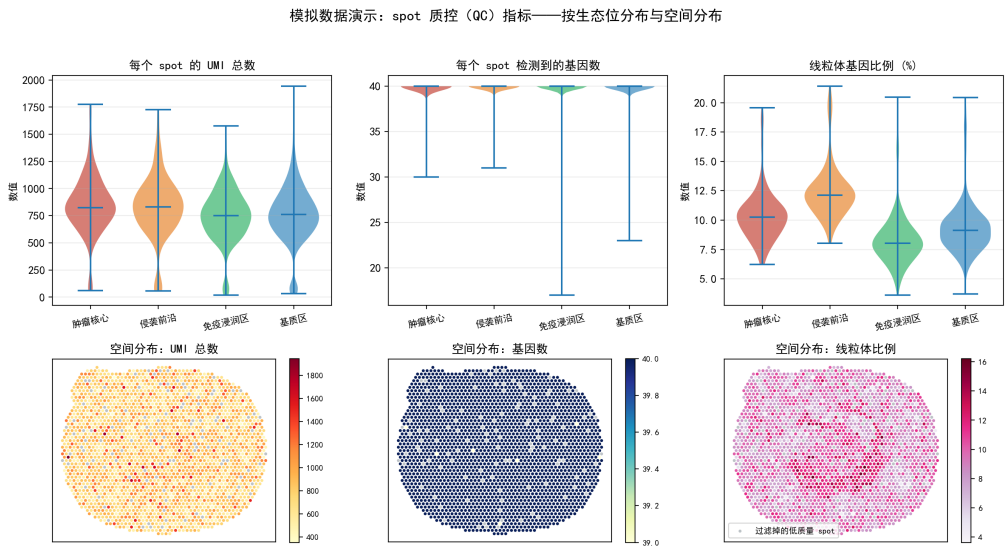

图 11: 图 7-1 本书示例数据的质控总览（模拟数据演示）：上方为按生态位的 UMI 数/基因数/线粒体比例分布，下方为对应指标的空间散点图。第 8 章将详细讲解如何读懂与产出这类图。

校验值后跑分析。

- 目录约定：data/raw 只读原始数据，data/processed 放产物，data/ 不进版本库。

练习与思考

1. **动手题**：完成 7.7 的下载与登记，把 data/README.md 的登记表填满（含 md5），并提交 data/samples.csv。
2. **判断题**：以下说法对错？（a）只下载 H5 不下载 spatial 目录也能做空间分析；（b）data/ 目录应该提交进 git 方便备份；（c）md5 校验值的作用是判断文件是否损坏。
3. **选择题**：你想在“有金标准手工注释”的数据上练习生态位识别，优先用哪个渠道？（a）10x 官网；（b）spatialLIBD；（c）Zenodo 任意搜索。
4. **思考题**：为什么“先记校验值、后跑分析”？如果数据在分析中途损坏，你如何发现？
5. **讨论题**：真实课题中你想用 GEO 上的某个乳腺癌 Visium 队列，登记表里除了 URL 还需要记什么？请列出至少三条你会在论文方法节写明的信息。

第 08 章预处理与质控

本章配套脚本：code/03_preprocess.R；本章图：figures/fig07_qc.png、figures/fig08_clustering.png

拿到 Visium 数据后，你手里是一张 “spot × 基因” 的计数矩阵，外加每个 spot 的组织坐标。直接分析它是个坏主意：捕获过程中混进了背景 RNA、低质量 spot，文库深度也参差不齐。预处理与质控（quality control, QC）就是把脏数据清洗成可靠数据的第一步，也是后面所有分析的地基。地基不稳，后面的聚类、注释、通讯推断全都白搭。这一章我们手把手把这一步做完。

本章目标

1. 学会用 Load10X_Spatial 读入 Visium 数据并理解它的目录结构。
2. 理解三个核心 QC 指标：nCount_Spatial（UMI 总数）、nFeature_Spatial（基因数）、percent.mt（线粒体比例）。
3. 掌握 “先看分布、再定阈值” 的过滤方法，而不是拍脑袋。
4. 学会归一化（SCTransform）、高变基因、PCA、聚类、UMAP 的完整流程。
5. 会用 SpatialDimPlot / SpatialFeaturePlot 做空间可视化，并正确解读 QC 图与聚类图。

8.1 读入数据：Load10X_Spatial

第 07 章我们从 10x Genomics 官网下载了人乳腺癌 demo 数据。下载完成后，data/visium/ 目录里应该包含这些内容：

- filtered_feature_bc_matrix.h5：spot × 基因的计数矩阵（HDF5 格式，包含基因名、spot 条形码）；
- spatial/ 目录：tissue_positions_list.csv（每个 spot 的组织坐标）、scalefactors_json.json（图像缩放因子）、tissue_hires_image.png 与 tissue_lowres_image.png（组织切片染色图）。

Load10X_Spatial 会把这些信息整合成一个 Seurat 对象。注意两个参数：data.dir 指向包含 spatial/ 子目录的文件夹（而不是 h5 文件本身）；filename 是 h5 文件的文件名。第一个代

码块先安装并加载 Seurat v5：

```
# 若未安装：install.packages("Seurat") 或 BiocManager::install("Seurat")
library(Seurat)
library(dplyr)      # 管道操作 %>%

# 读入 Visium 数据：data.dir 必须指向含 spatial/ 子目录的文件夹
obj <- Load10X_Spatial(
  data.dir  = "data/visium/",
  filename  = "filtered_feature_bc_matrix.h5"
)

# 看一眼对象结构：应该显示 1 个 assay（Spatial）和约 5000 个 spot
obj
```

运行结果解读：终端会打印类似 `15000 features across 4780 samples within 1 assay` 的信息。“features” 是基因数，“samples” 在这里就是 spot 数（10x Visium 捕获区约 5000 个 spot，过滤后通常剩 4000–5000 个）。Seurat v5 会自动把 assay 命名为 `Spatial`，图像信息存在 `obj@images` 里，坐标存在 `obj@images$slice1@coordinates`。此时 `obj` 里还没有任何质控指标，下一步来算。

8.2 QC 指标的含义：先看懂三个数字

每个 spot 是一个直径约 55 μm 的小圆点，覆盖约 1–10 个细胞（见第 02 章）。由于捕获和建库过程的随机性，每个 spot 的质量参差不齐。我们关心三个指标：

① nCount_Spatial：每个 spot 捕获到的 UMI 总数（唯一分子标识符，unique molecular identifier）。UMI 是建库时给每条 mRNA 分子贴的“身份证号”，所以 UMI 总数反映该 spot 的文库深度（测到了多少转录本）。太低说明这个 spot 可能没捕获到组织（背景 spot 或空 spot），太高则要警惕是不是技术异常。

② nFeature_Spatial：每个 spot 检测到的基因种类数。它与 UMI 数正相关（测得多自然基因多），但两者并不完全等价——一个 spot 如果只测到寥寥几个基因，说明它信息量太少，无法支撑后续分析。

③ percent.mt：线粒体基因（mitochondrial genes）占总 UMI 的比例。在单细胞分析里，线粒体比例高常提示细胞破裂、胞浆 RNA 流失，只剩线粒体 RNA 残留，代表坏死或濒死细胞。在 Visium 里同样适用：组织边缘或坏死区域的 spot 往往 percent.mt 偏高。这个指标需要我们自己算，Seurat 提供了现成函数：

```
# 计算每个 spot 的线粒体基因比例（人样本线粒体基因以 MT- 开头）
obj[["percent.mt"]] <- PercentageFeatureSet(obj, pattern = "^MT-")
```

```
# 查看前 6 个 spot 的质控指标（行是 spot，列是指标）
head(obj[[]])
```

运行结果解读：`obj[[]]` 是 Seurat 对象的元数据（metadata）表，现在包含 `orig.ident`、`nCount_Spatial`、`nFeature_Spatial`、`percent.mt` 四列。`head()` 会打印前 6 行，你可以看到不同 spot 之间数字差异很大：有的 spot UMI 数上万，有的只有几百。这就是为什么必须做归一化和过滤。

> **为什么模式是"^MT-"？**人线粒体基因名以 `MT-` 开头（如 MT-ND1、MT-CO1），正则表达式 `^MT-` 表示"以 MT- 开头"。小鼠数据要用**"^mt-"**，分析前先确认物种，别搞混。

8.3 阈值设定：先看分布，不拍脑袋

新手最常见的错误是直接抄别人论文的阈值（比如"基因数 > 200 就留下"）。不同组织、不同建库批次的数据分布天差地别，抄来的阈值可能把好 spot 全过滤掉，也可能留下大片垃圾。正确做法分三步：**看分布→按低分位数粗定→结合生物学常识微调**。

```
# 第一步：画分布，直观感受数据长什么样
# VlnPlot 是 Seurat 的默认"体检"图，横轴是 spot 分组（未分组时全是 sample1）
VlnPlot(obj, features = c("nCount_Spatial", "nFeature_Spatial", "percent.mt"),
        ncol = 3, pt.size = 0)
```

```
# 直方图补充：更精确地看数量分布（用 ggplot2 画 nCount 的直方图）
library(ggplot2)
ggplot(obj[[]], aes(x = nCount_Spatial)) +
  geom_histogram(bins = 80, fill = "#5B9BD5", color = "white") +
  labs(x = "UMI 总数（nCount_Spatial）", y = "spot 数量", title = " 文库深度分布")
```

运行结果解读：VlnPlot 会画出三个小提琴图，小提琴的"肚子"显示大多数 spot 落在哪个范围。直方图则更清楚：如果绝大多数 spot 的 UMI 数集中在 1000–8000，只有极少数 spot 接近 0，那接近 0 的那一小撮就是可疑对象。图 8-1 上排就是"先看分布"这一步的成品（见 8.7 节）。

第二步：定阈值。一个实用经验是取分布的低分位数（比如 5%–10%）作为下限，再结合组织类型常识微调。对 Visium 人乳腺癌 demo，教程统一采用下面的阈值（与第 00-TECH 速查第 3 节一致）：

```r
# 过滤：同时满足 基因数 >200 且 UMI 数 >500 且 线粒体比例 <25%
obj <- subset(obj, subset = nFeature_Spatial > 200 &
                      nCount_Spatial > 500 & percent.mt < 25)

# 过滤后还剩多少 spot？拿过滤前的数字对比一下
ncol(obj)
```

运行结果解读：`ncol(obj)` 返回剩余的 spot 数量。以 10x 乳腺癌 demo 为例，通常从约 4700 个 spot 过滤到约 4400–4600 个，丢掉 2%–6% 的异常 spot。注意三个条件用 `&`（且）连接，表示“任何一个指标不合格就淘汰”——这是更严格的策略；如果你只想剔除“极端异常”的 spot，可以放宽阈值，让更多 spot 留在下游分析里，之后再验证它们是否带来噪声。

第三步：验证过滤效果。过滤后再画一次 VlnPlot，确认留下的 spot 分布干净、没有拖尾。这一步很多人跳过，但它其实是最快的自检。

> **阈值怎么微调？**判据：过滤后 spot 数量不应骤降（骤降说明阈值太狠，好数据被误杀）；低质量 spot 应主要位于组织边缘或组织外（对照 tissue 图像看，见 8.6 节）。这两个判据比“抄论文数字”靠谱得多。8.8 节的练习让你亲手体验阈值变化的影响。

8.4 归一化：LogNormalize 还是 SCTransform？

即使过滤干净，不同 spot 的文库深度仍然不同：一个 spot 测到 8000 个 UMI，另一个只测到 2000 个，前者每个基因的计数天然更大。如果不做归一化，聚类时“文库深度差异”会伪装成“表达差异”，把 spot 按深度而不是按细胞类型分开。归一化的目标就是消除这种技术差异。

Seurat 提供两条主流路线：

方法	做法	优点	缺点
LogNormalize	每个基因的计数除以该 spot 总 UMI，乘 10000，再取 log1p（`log(x+1)`）	简单、快、兼容性好	两步走（归一化后再单独找高变基因）；对文库深度校正不够精细
SCTransform	用正则化负二项回归同时建模 UMI 深度等技术因素与生物学异质性，一步完成归一化 + 高变基因选择	校正更彻底，聚类更干净，是当前推荐默认	明显更慢、更吃内存

选型建议：本书统一用 SCTransform（更优）；只有当你的机器跑不动（见第 06 章内存提示：Visium 单切片约 2–6 GB），或你明确需要与旧脚本的 LogNormalize 结果对齐时，才退回 `NormalizeData` + `FindVariableFeatures`。SCTransform 的输出直接替代“归一化 + 高变基因”两步，所以第 8.5 节里不需要再单独找高变基因：

```
# SCTransform：一步完成归一化与高变基因选择（自动把结果存进 SCT assay）
obj <- SCTransform(obj, assay = "Spatial", verbose = FALSE)
```

运行结果解读：运行后 `obj` 里会出现一个名为 `SCT` 的新 assay。之后 PCA、聚类、UMAP 默认都用 `SCT` 的数据。`verbose = FALSE` 只是关掉进度刷屏，不影响结果。

> **对比记忆**：`LogNormalize` 相当于“除以总量再取对数”，`SCTransform` 相当于“对每个基因拟合一个带文库深度协变量的回归，取残差”。后者更精细，代价是计算量。

8.5 高变基因、PCA 与聚类

归一化之后，标准流程是：**降维→建图→聚类→再降维可视化**。

- **PCA（主成分分析，principal component analysis）**：把上万维的基因空间压缩成几十个“主成分”，每个主成分是基因的线性组合，捕捉最大的表达变异。我们取前 30 个主成分，因为前几个主成分包含绝大部分信息，后面基本是噪声。
- **FindNeighbors**：基于 PCA 空间为每个 spot 找邻居，构建 KNN 图（K 近邻图）。
- **FindClusters**：在 KNN 图上做 Louvain 社区检测，把图分成若干“表达相似”的群落——这就是聚类（cluster）。
- **RunUMAP**：把高维数据压到 2 维，方便肉眼观察聚类结构（UMAP 只用于可视化，聚类本身基于 PCA 空间）。

```
# 完整降维聚类管道（与 00-TECH 第 3 节一致）
obj <- RunPCA(obj) %>%
  FindNeighbors(dims = 1:30) %>%
  FindClusters(resolution = 0.6) %>%
  RunUMAP(dims = 1:30)

# 看 UMAP 散点：每个点一个 spot，按聚类着色
DimPlot(obj, reduction = "umap", label = TRUE)
```

运行结果解读：`DimPlot` 会画出 UMAP 散点图，通常能看到几个抱团的群落，每个团一个颜色、一个 cluster 编号（如 0–7）。`resolution = 0.6` 控制聚类“颗粒度”：数值越大聚类越多越碎，越小越少越粗。0.6 是 Visium 默认推荐值，先跑一遍看结果再调。聚类编号本身没有生物学含义——0 号不等于“最好”，它只是社区检测的序号，**注释**是第 09 章的事。

8.6 空间可视化入门

单细胞分析里我们看 UMAP，空间转录组里更关键的是**把结果映射回组织切片**，看它长在哪个位置。Seurat 提供两个专属函数：

- `SpatialDimPlot`：按聚类/分组着色，画在组织图像上；
- `SpatialFeaturePlot`：按某个基因的表达量着色，画在组织图像上。

```
# 聚类结果映射回组织切片：颜色相同的 spot 属于同一个 cluster
SpatialDimPlot(obj, label = TRUE)

# 单个基因的空间表达：看 EPCAM（上皮/肿瘤 marker）长在哪
SpatialFeaturePlot(obj, features = "EPCAM", pt.size.factor = 1.2)
```

运行结果解读：`SpatialDimPlot` 输出的是一张组织切片图，每个 spot 一个色块，能直观看到哪些 cluster 聚在组织中央、哪些贴着边缘——这是后面识别“空间生态位”（第 10 章）的雏形。`SpatialFeaturePlot` 用颜色深浅表示表达量：EPCAM 高表达的区域（深色）大概率是肿瘤细胞聚集区。看到这里你应该能体会到空间转录组的魅力：UMAP 丢掉的“位置信息”在这里回来了。

8.7 结果解读：QC 图与聚类图

现在把本章的产出图放在一起看。图 8-1 是 QC 三联图（模拟数据演示）：

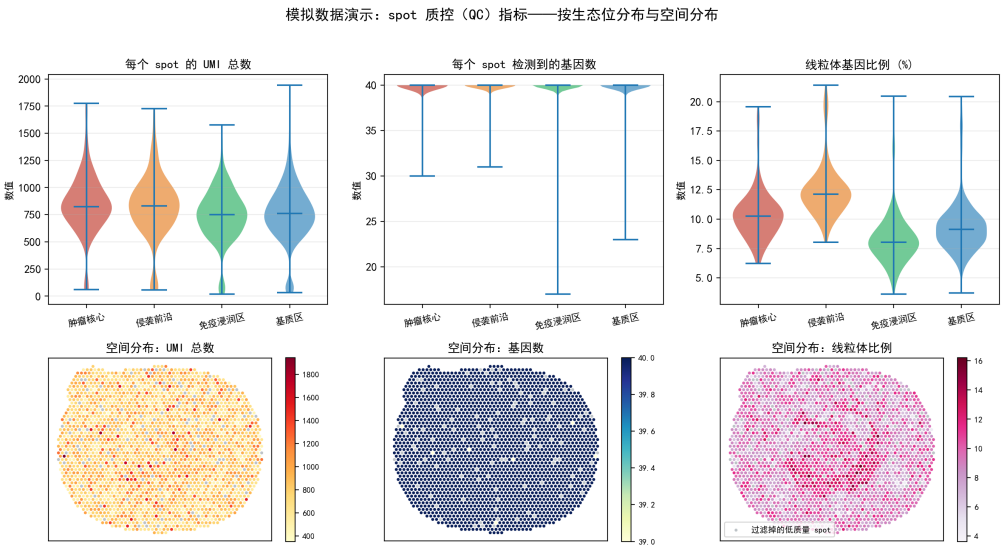

图 12：图 8-1 质控指标分布（模拟数据演示）。上排：按空间生态位的 UMI 数、基因数、线粒体比例小提琴图；下排：三个指标映射回组织切片的空间散点图，低质量 spot 用灰色标出。

怎么读这张图：上排三个小提琴图告诉你指标的整体分布——小提琴越“胖”的地方 spot 越多。如果线粒体比例的小提琴在 25% 以上还有一大截，说明坏死组织占比高，过滤阈值要收

紧；如果小提琴整体贴着 0，说明数据干净，阈值可以放宽。下排空间散点图是“位置版”的指标：正常组织里，低 UMI、低基因、高线粒体的 spot（灰色）应该零星散布在组织边缘或组织外——如果它们成片出现在组织中央，那更可能是切片或捕获环节出了问题，而不是单纯的质量差。**注意：本图为模拟数据演示，仅用于展示“真实分析会长什么样”，不代表任何真实样本的结论。**

图 8-2 是降维聚类结果（模拟数据演示）：

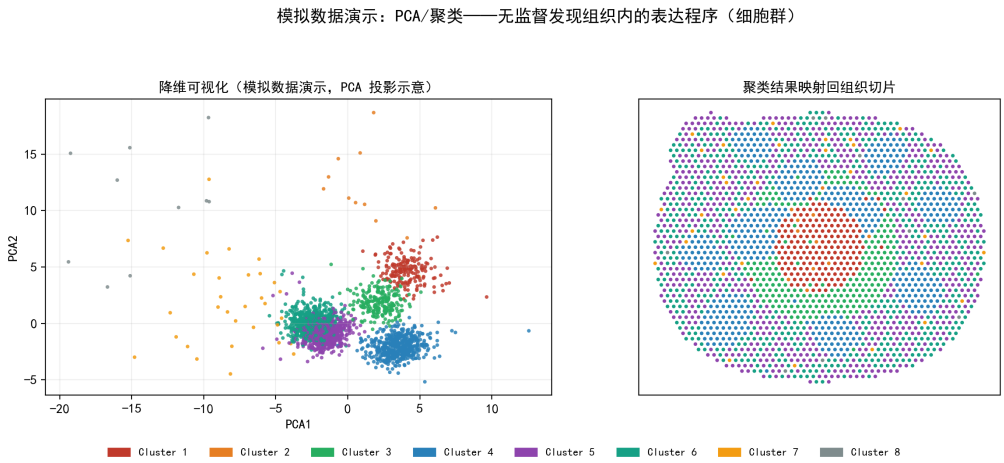

图 13: 图 8-2 聚类结果（模拟数据演示）。左：UMAP 降维散点图，共 8 个 cluster；右：同一聚类结果映射回组织切片的空间分布。

怎么读这张图：左图是“无位置视角”——8 个 cluster 在 UMAP 上各自成团，说明它们的表达谱确实不同；右图是“有位置视角”——可以看到某些 cluster 的空间位置高度集中（比如一大片红色占据切片中央），这通常对应组织结构里的某一类区域。**注意：**聚类是纯数据驱动（只看表达，不看位置），它和后面 BayesSpace 的“空间聚类”（第 10 章，把位置信息也纳入）是两回事，别混淆。此时我们还不知道每个 cluster 是什么细胞——注释是第 09 章的任务。

8.8 动手练习：调整阈值，观察 spot 数量变化

过滤阈值没有标准答案，最好的老师是自己动手。做下面这个实验：

```
# 练习：先记住过滤前的 spot 数量
n_before <- ncol(obj)

# 尝试三组阈值，分别记录剩余的 spot 数
thr1 <- subset(obj, subset = nFeature_Spatial > 200 &
                      nCount_Spatial > 500 & percent.mt < 25)
thr2 <- subset(obj, subset = nFeature_Spatial > 500 &
                      nCount_Spatial > 1000 & percent.mt < 10)
thr3 <- subset(obj, subset = nFeature_Spatial > 100 &
                      nCount_Spatial > 200 & percent.mt < 40)
```

```
# 打印对比结果（用英文列名，避免中文列名在部分环境被改写）
data.frame(
  plan    = c("过滤前", "宽松 (thr3)", "教程默认", "严格 (thr2)"),
  n_spots = c(n_before, ncol(thr3), ncol(thr1), ncol(thr2))
)
```

运行结果解读：你会看到“严格”方案可能丢掉 30% 以上的 spot，“宽松”方案几乎不丢。接着分别对 `thr2` 和 `thr3` 跑一次 8.5 节的聚类管道，对比 UMAP 和 `SpatialDimPlot`：如果“宽松”方案出现一个主要由低质量 spot 组成的小团，说明阈值太松；如果“严格”方案把组织边缘的有用信息也删了，说明阈值太狠。**练习的结论不是“哪个阈值对”，而是让你建立“阈值与下游结果联动”的感觉**——这正是做科研和跑脚本的区别。

本章小结

本章完成了空间转录组分析的第一块地基：用 `Load10X_Spatial` 读入数据，计算并理解三个 QC 指标（UMI 总数、基因数、线粒体比例），按“先看分布、再定阈值”的原则过滤低质量 spot，用 SCTransform 归一化，跑通 PCA → FindNeighbors → FindClusters → RunUMAP 的完整管道，最后学会把结果映射回组织切片。现在你手里有一个干净、归一化、带聚类的 Seurat 对象，下一步（第 09 章）就是回答最关键的问题：**每个 cluster、每个 spot 里到底是什么细胞？**

练习与思考

1. **概念题：**为什么 `nCount_Spatial` 低的 spot 要过滤掉？如果不过滤，它会对后续聚类造成什么影响？
2. **概念题：**线粒体比例高的 spot 在生物学上通常意味着什么？这个指标在 Visium 和单细胞（scRNA-seq）里的解读有什么异同？
3. **动手题：**按 8.8 节的三组阈值分别跑一遍过滤，记录 spot 数量变化，并解释“严格”与“宽松”两种策略各自的利弊。
4. **动手题：**对过滤后的对象画 `SpatialFeaturePlot` 展示基因 `EPCAM` 和 `COL1A1`，观察两者空间分布是否不同，想想这提示了什么（提示：与乳腺癌的组织结构有关）。
5. **思考题：**SCTransform 相比 LogNormalize 多做了什么？在什么情况下你会选择更慢的 SCTransform？（提示：从聚类稳定性角度思考。）

第 09 章降维聚类与细胞类型注释

本章配套脚本：`code/04_annotation.R`（含 cell2location 调用说明）；本章图：`figures/fig09_celltype.png`

第 08 章结束，我们有了一个干净、归一化、带聚类的 Seurat 对象。但聚类编号（0、1、2……）本身没有生物学含义——它是 “表达相似的 spot 群落”。本章要回答的问题只有一个：**每个 cluster 里到底是什么细胞？**这一步叫细胞类型注释（cell type annotation）。它是承上启下的一环：没有注释，第 10 章的生态位无法命名，第 11 章的通讯分析没有 “说话的主体”。本章给你三套策略，从最轻量到最严谨，并教你遇到问题怎么排查。

本章目标

1. 理解为什么 Visium 的 spot 必须注释：每个 spot 是 1–10 个细胞的混合信号，不是单细胞。
2. 掌握策略 A：用 marker 基因（标志基因）直接注释，会看 marker 的空间分布。
3. 理解策略 B：单细胞参考去卷积（deconvolution）的原理，会用 cell2location 估计每个 spot 的细胞类型丰度。
4. 了解策略 C：与公共单细胞图谱整合注释。
5. 能正确解读注释结果图（fig09），并处理 “marker 不特异” “与病理不符” 等常见问题。

9.1 为什么 spot 需要注释：混合信号的真相

先想清楚一个底层事实：**Visium 的 spot 不是单细胞**。每个 spot 直径约 55 μ m，覆盖约 1–10 个细胞（第 02 章）。因此一个 spot 的计数矩阵是多种细胞表达的叠加——比如一个 spot 里同时有 3 个肿瘤细胞和 2 个巨噬细胞，它的表达谱就是 “肿瘤 + 巨噬” 的混合体。

这带来两个推论：

1. **聚类≠细胞类型**。单细胞数据里一个 cluster 通常≈一种细胞；Visium 里一个 cluster 可能是 “某种细胞占主导” 的区域，也可能是几种细胞的稳定组合。
2. **注释的对象是 “spot 的主要成分” 或 “细胞类型比例”**，而不是 “这个 spot 就是某某细胞”。所以注释策略天然分两派：直接派（这个 spot 更像谁）和比例派（这个 spot 里各占多少），本章的 A/B/C 三套策略正是这两派的体现。

理解了这一点，你就不会在看到“巨噬细胞 spot”时误以为那个 spot 里只有巨噬细胞。

9.2 策略 A：marker 基因直接注释（最轻量，先做它）

marker 基因（标志基因）是某类细胞特征性高表达、其他细胞低表达的基因。乳腺癌 Visium 的常见 cell type 与 marker 如下（与 00-TECH 速查第 7 节完全一致，全书统一）：

细胞类型	marker 基因	备注
肿瘤细胞（上皮）	EPCAM / KRT8 / KRT19	上皮细胞标志，肿瘤组织里通常是癌细胞
癌症相关成纤维细胞 CAF	COL1A1 / COL3A1 / DCN / PDGFRA	成纤维细胞活化后高表达胶原与 PDGFRA
CD8 T 细胞	CD3D / CD8A / GZMB	CD3D 是所有 T 细胞的通用标志，CD8A/GZMB 定位到杀伤性 T
巨噬细胞	CD68 / CD163 / C1QA	肿瘤相关巨噬细胞（TAM）的标志
B 细胞	MS4A1 / CD79A	淋巴细胞标志
内皮细胞	PECAM1 / VWF	血管内皮标志

直接注释的操作流程：**把每个 marker 基因画到组织切片上（SpatialFeaturePlot）→肉眼观察哪些区域高表达→给 cluster 或 spot 打分→命名**。先看空间分布：

```
# 把 6 类细胞的代表 marker 画到组织切片上，一次看 6 张
SpatialFeaturePlot(obj,
  features = c("EPCAM", "COL1A1", "CD3D", "CD68", "MS4A1", "PECAM1"),
  ncol = 3, pt.size.factor = 1.2)
```

运行结果解读：你会看到 6 张切片图。理想情况下：EPCAM 高表达区域成片出现在肿瘤主体；COL1A1 环绕在肿瘤周围（基质）；CD3D/CD68 呈散点状散布（免疫浸润）；PECAM1 呈细条状网络（血管）。**如果某个 marker 全片低表达，先别慌**，可能是数据问题也可能是真实的生物学（见 9.6 节）。

肉眼观察之后，把观察变成可重复的“打分”：用 `AddModuleScore` 把同一细胞类型的多个 marker 合成一个模块分数，再按分数给每个 spot 判断主要细胞类型。

```
# 定义 6 类细胞的 marker 列表（与上方表格一致）
marker_list <- list(
  Tumor   = c("EPCAM", "KRT8", "KRT19"),
```

```
  CAF     = c("COL1A1", "COL3A1", "DCN", "PDGFRA"),
  CD8T    = c("CD3D", "CD8A", "GZMB"),
  Macro   = c("CD68", "CD163", "C1QA"),
  Bcell   = c("MS4A1", "CD79A"),
  Endo    = c("PECAM1", "VWF")
)

# 每个 spot 计算 6 个模块分数（列名自动为 score1~score6，与 marker_list 顺序对应）
obj <- AddModuleScore(obj, features = marker_list, name = "score")

# 每个 spot 取分数最高的细胞类型作为"主要类型"
score_cols <- paste0("score", 1:length(marker_list))
obj$celltype <- names(marker_list)[apply(obj[[]][score_cols], 1, which.max)]

# 看看各类型 spot 数量
table(obj$celltype)

# 主要细胞类型画到切片上
SpatialDimPlot(obj, group.by = "celltype", label = TRUE)
```

运行结果解读：`table()` 给出每种主要类型的 spot 数——如果"CD8T"占了 60%，八成有问题（T 细胞不可能占乳腺癌切片的一半），需要检查打分逻辑或回到 marker 分布图。`SpatialDimPlot` 把"每个 spot 的主要细胞类型"映射回切片，你会看到肿瘤区域一片 Tumor 色、基质区域一片 CAF 色——这就是策略 A 的最终产物。

策略 A 的适用场景：快速出图、初步判断、以及作为策略 B/C 的交叉验证。它的局限也很明显：只回答"谁占主导"，答不了"比例是多少"，而且对混合严重的 spot 容易误判。

9.3 策略 B：单细胞参考去卷积（更定量）

去卷积（deconvolution）解决策略 A 回答不了的问题：**每个 spot 里各种细胞类型各占多少**。思路是：先有一个参考——高质量的同一组织类型单细胞图谱（已知每个细胞是什么类型）——然后反推：把每个 spot 的混合表达谱拆解成各细胞类型的比例。

cell2location 原理（本教程主推）：它分两步走，都是贝叶斯模型。

- **第一步：**用参考单细胞数据训练一个负二项回归模型，学到每种细胞类型的"平均表达谱"与基因离散度参数——相当于给每种细胞建一份"基因指纹"。
- **第二步：**把第一步学到的指纹固定住，用来解释 Visium 每个 spot 的总 UMI：模型假设 spot 的计数 = 若干种细胞类型的指纹按"每种类型有多少个细胞"加权求和，然后用变分推断（variational inference）反推出每个 spot 每种细胞类型的**绝对细胞数**（丰度）。

输出的核心矩阵是 spot × 细胞类型的丰度表，存于 `adata.obsm["q05_cell_abundance_w_sf"]`。

下面是核心代码（Python，cell2location ≥ 0.8；需要先准备好参考单细胞 h5ad 与 Visium h5ad，见第 07 章数据清单）：

```
import cell2location
from cell2location.models import RegressionModel, Cell2location
import scanpy as sc

# 输入准备：参考单细胞图谱（含细胞类型标签列 "cell_type"）与 Visium 数据
adata_ref  = sc.read_h5ad("data/ref_breast_sc.h5ad")    # 参考单细胞
adata_vis  = sc.read_h5ad("data/visium_breast.h5ad")    # Visium（scanpy 版）

# 第一步：用参考单细胞训练"指纹"模型
ref_model = RegressionModel(adata_ref)
ref_model.train(max_epochs=250, use_gpu=False)
adata_ref = ref_model.export_posterior(adata_ref)

# 第二步：固定指纹，估计每个 spot 的细胞类型丰度
# N_cells_per_location：每个 spot 平均细胞数，Visium 取 10（覆盖 1-10 个细胞）
mod = Cell2location(adata_vis, adata_ref, N_cells_per_location=10)
mod.train(max_epochs=30000, use_gpu=False)

# 导出丰度矩阵并写回 adata
adata_vis = mod.export_posterior(adata_vis)
abundance = adata_vis.obsm["q05_cell_abundance_w_sf"]   # spot × 细胞类型
abundance.head()
```

运行结果解读：`abundance` 是数据框，行是 spot，列是细胞类型，数值是"该 spot 里大约有几个这种细胞"。把丰度最高的类型作为 spot 的主要细胞类型，可以画出和策略 A 类似的切片图；把丰度按生态位汇总，就是 9.5 节堆叠条形图的原料。注意两点：训练要跑很久（教程里 `max_epochs=30000` 是保守设置，实际常用早停，机器建议≥ 16 GB 内存，见第 06 章）；`use_gpu=False` 表示纯 CPU 也能跑，只是慢。

一句话认识另外两个去卷积工具：RCTD（R 包 spacexr）用受限二次规划把参考表达谱拟合到 spot 上，速度快、文档友好；SPOTlight 先用非负矩阵分解（NMF）从参考数据提炼"细胞类型特征基"，再用非负最小二乘估计比例。三者结论通常大体一致；cell2location 的卖点是能估计**绝对细胞数**（而非只给相对比例），更适合我们后面"按生态位比较组成"的需求。

9.4 策略 C：与公共单细胞图谱整合注释

如果你手头没有自己的单细胞参考，可以用公共图谱。乳腺癌方向有多个公开的单细胞图谱（如 Broad 的 single cell portal、CellxGene 上的乳腺癌图谱，见第 07 章数据来源）。整合注释的通用做法是 Seurat 的标签转移（label transfer）：把 Visium 和参考图谱放进同一个低维空间，让参考的细胞类型标签"传导"到空间 spot 上。

```
# 伪代码：Seurat 标签转移流程（参考图谱 adata_ref 需先转成 Seurat 对象）
# anchors <- FindTransferAnchors(reference = ref, query = obj, dims = 1:30)
# pred <- TransferData(anchorset = anchors, refdata = ref$cell_type, dims = 1:30)
# obj$celltype_integrated <- pred$predicted.id
```

上面是伪代码：真实运行需要先下载并整理参考图谱（第 07 章的下载清单）。本书以策略 A/B 为主，策略 C 的价值在于**没有自备参考时的备选**，以及多策略交叉验证。

三套策略怎么选？实践建议：先用策略 A 快速出图建立直觉→有条件就用策略 B 拿到定量比例（本课题主线）→用策略 C 或文献结果交叉验证，三者一致时结论才敢写进论文。

9.5 结果解读：注释结果图

图 9-1 是注释结果示例（模拟数据演示）：

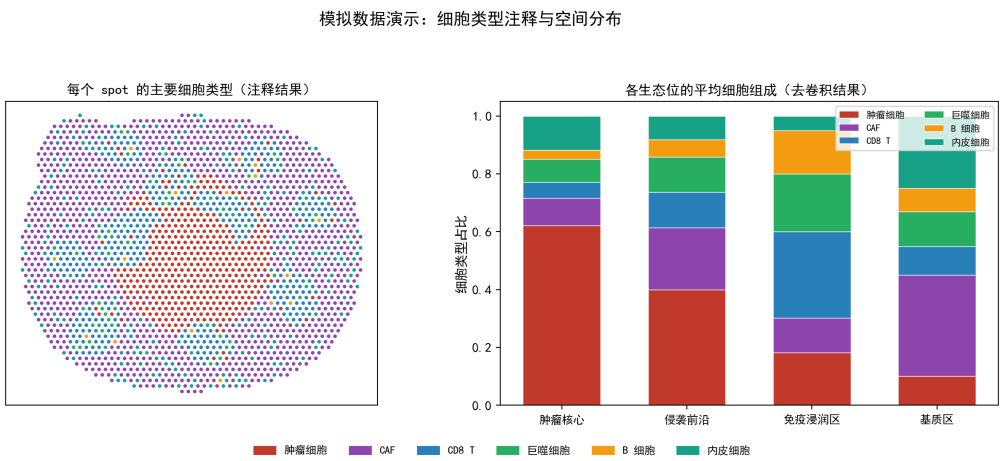

图 14: 图 9-1 细胞类型注释结果（模拟数据演示）。左：每个 spot 的主要细胞类型映射回组织切片；右：按空间生态位汇总的平均细胞组成堆叠条形图。

怎么读这张图：左图看“空间”——理想结果里，Tumor 色块成片位于切片中央，CAF 色块包裹在外周，免疫细胞（CD8T、Macro、Bcell）散布其间，Endo 呈细线穿插——这与乳腺癌“肿瘤巢 + 间质 + 免疫浸润”的组织学结构相符。右图看“组成”——每个生态位（柱）里各细胞类型的占比（堆叠色块），一眼就能比较“免疫浸润区”和“肿瘤核心”的细胞构成差异。**注意：本图为模拟数据演示，仅用于展示“真实分析会长什么样”。**右图的生态位划分来自第 10 章，这里先眼熟它——下一章我们就要亲手做出来。

统计提示：如果后续要对生态位间的细胞组成做差异检验（比如“侵袭前沿的巨噬细胞比例显著更高吗”），建议用置换检验（permutation test）或广义线性模型，并对多重比较做 BH-FDR 校正（见第 00-SPEC 第 8 节），不要直接用 `t.test` 反复比。

9.6 常见问题排查

问题 1：marker 不特异怎么办？

这是新手注释翻车的第一大原因。比如 `CD68` 在巨噬细胞和树突状细胞里都表达；`EPCAM` 在发生上皮间质转化（EMT）的肿瘤细胞里会下调，导致“肿瘤区域测不到 EPCAM”。对策：

- 用**多个 marker 联合打分**（策略 A 的 `AddModuleScore` 已经这么做了），单个 marker 高不算数；
- 换更特异的 marker（如巨噬细胞加看 `CD163`、`MRC1`；T 细胞加看 `CD4`/`CD8A` 细分）；
- 用去卷积结果交叉验证：如果 cell2location 说某个区域巨噬细胞丰度很高，而 marker 打分却低，多半是 marker 选择问题而不是数据问题；
- 查阅文献：以“乳腺癌 + 空间转录组 + marker”为关键词检索（见第 07 章文献检索记录），用同行验证过的 marker 列表。

问题 2：注释结果与病理（H&E）不符怎么办？

比如病理切片上明明是大片肿瘤，注释却标成基质。可能的原因按可能性排序：

1. **EMT 或异质性**：部分肿瘤细胞下调上皮 marker，看起来像基质（生物学真相，不是 bug）；
2. **混合信号稀释**：肿瘤细胞与基质细胞共处一个 spot，marker 信号被稀释到阈值以下（Visium 分辨率的固有限制）；
3. **参考不匹配**：策略 B/C 的参考图谱来自不同亚型或不同平台，批次效应污染了注释。

排查顺序建议：先回到原始表达（`SpatialFeaturePlot` 直接看 marker），再换一套 marker 或换策略，最后才考虑“是不是数据读错了”。**切记：注释是计算推断，最终解释权在病理与实验证据**，教程的注释只能作为假说来源。

9.7 动手练习

1. 用策略 A 的 `AddModuleScore` 流程注释你的对象，把 `table(obj$celltype)` 的结果与 9.2 节的示例对照，解释各类 spot 数量的合理性。
2. 把 cell2location 的 `abundance` 结果按“丰度最高的类型”画成切片图，与策略 A 的结果叠加比较：哪些 spot 两种策略一致？哪些不一致？试着解释不一致的原因。
3.（进阶）下载一个公共乳腺癌单细胞图谱，用 9.4 节的标签转移流程注释，与策略 A/B 三方交叉验证。

本章小结

本章回答了 “spot 里是什么细胞”：先讲清 spot 是 1–10 个细胞的混合信号这个底层事实，然后给出三套注释策略——marker 直接注释（快、直观）、cell2location 去卷积（定量、给出比例）、公共图谱整合（无参考时的备选与验证）。同时教了两种常见问题的排查思路。注释完成后，每个 spot 都有了 “主要细胞类型” 或 “丰度向量”，这正是下一章识别空间生态位的输入——因为生态位的本质，就是 “细胞组成有特色、空间位置连成片” 的区域。

练习与思考

1. **概念题**：为什么说 “Visium 聚类得到的 cluster ≠细胞类型”？混合信号对注释策略的选择有什么影响？
2. **概念题**：cell2location 与策略 A 的根本区别是什么？（提示：一个回答 “谁占主导”，一个回答 “各占多少”）
3. **动手题**：对 `EPCAM`、`KRT8`、`KRT19` 三个 marker 分别画 `SpatialFeaturePlot`，观察它们的高表达区域是否完全重合。如果不重合，说明了什么？
4. **动手题**：把 cell2location 的丰度结果与策略 A 的主要类型结果做一个一致性表格（如列联表），统计一致率，并讨论不一致 spot 的特征。
5. **思考题**：如果你的数据里 B 细胞 marker（MS4A1/CD79A）全片几乎不表达，你会先怀疑数据质量，还是先怀疑生物学？请列出你验证判断的具体步骤。

第 10 章空间生态位识别

本章配套脚本：`code/05_spatial_domain.R`；本章图：`figures/fig10_niche.png`、`figures/fig11_nhood.png`

前面两章我们完成了“每个 spot 是什么”的拼图：第 08 章清洗数据，第 09 章注释细胞类型。但从“spot 有细胞类型”到“组织里有几个生态位”，还差关键一步——**把散落的 spot 组织成有生物学意义的空间区域**。这就是本章的主题：空间生态位识别（spatial niche identification）。它是本课题的核心章节：没有生态位，就没有“分区通讯”，也就没有后面的“通讯重编程”故事。本章的三步法会带你从分割、验证到命名，完整走一遍。

本章目标

1. 理解生态位识别“三步法”的整体框架：空间域分割→邻域富集验证→命名与特征化。
2. 学会用 BayesSpace 做空间域分割，理解其马尔可夫随机场原理与聚类数 q 的选择。
3. 学会用 Squidpy 的邻域富集（nhood enrichment）验证细胞类型的空间共定位。
4. 掌握生态位命名规则与特征化方法（marker 基因 + Hallmark 通路富集）。
5. 能正确解读生态位结果图，并判断“几个生态位合适”。

10.1 三步法总览

生态位识别不是一个算法能完成的，而是一条验证链。本书采用三步法：

1. **空间域分割（spatial domain segmentation）**：用考虑空间位置信息的聚类算法，把相邻且表达相似的 spot 归为同一“空间域”。这一步产生“候选生态位”。
2. **邻域富集验证（neighborhood enrichment validation）**：用统计检验确认候选生态位的细胞组成在空间上确实“扎堆”——即某些细胞类型倾向于互为邻居，而不是随机散布。这一步验证分割结果不是统计假象。
3. **生态位命名与特征化（niche annotation & characterization）**：根据每个生态位的细胞组成与特征通路，结合组织学先验（比如肿瘤核心、侵袭前沿），给它们命名，并用差异表达与通路富集描述“这个生态位是什么、擅长干什么”。

三步缺一不可：第一步给地图，第二步验地图，第三步给地图写图例。下面依次实操。

10.2 第一步：空间域分割（BayesSpace 实操）

BayesSpace 原理

第 08 章的 Seurat 聚类只用了表达信息（PCA 空间），完全无视 spot 的物理位置，所以它可能把组织两侧“表达相似但位置不相邻”的 spot 聚到一起——这对识别空间区域是缺陷。BayesSpace 的做法是引入**空间先验**：它假设空间上相邻的 spot 更可能属于同一个空间域。具体地，BayesSpace 在聚类模型里加入一个**马尔可夫随机场（Markov random field, MRF）**，用一个“平滑”参数约束相邻 spot 的聚类标签趋于一致，再通过贝叶斯推断同时估计每个 spot 的域归属和基因表达模式。**聚类数 q 是唯一需要你拍板的关键参数**——它直接决定组织被切成几块。别担心，10.2 节和 10.6 节会教你怎么选。

实操代码

BayesSpace 的输入是 SingleCellExperiment（SCE）对象，所以第一步从 Seurat 转换（本教程以 q = 7 为例，参数与 00-TECH 速查第 3 节一致）：

```
library(BayesSpace)
library(SingleCellExperiment)

# 从 Seurat 对象转成 SCE（BayesSpace 的输入格式）
sce <- as.SingleCellExperiment(obj)

# 空间预处理：对数标准化 + 提取前 15 个主成分（供聚类使用）
sce <- spatialPreprocess(sce, platform = "Visium", n.PCs = 15)

# 选 q：用 qTune 对多个 q 值各跑一轮，比较结果（数据量大时较慢，可先用 4-8）
sce <- qTune(sce, qs = seq(4, 8), platform = "Visium", d = 15)
# qTune 会为每个 q 存一组聚类标签（colData 里的 q4~q8 列），
# 可用 clusterPlot(sce, label = "q4") 依次查看每个 q 的分割效果

# 正式空间聚类（此处 q 取自 qTune 的折中值，实践中按 10.6 节标准定）
sce <- spatialCluster(sce, q = 7, platform = "Visium", d = 15)

# 可视化：空间域映射回组织切片
clusterPlot(sce)

# 把空间聚类标签传回 Seurat 对象，便于后续统一操作
obj$BayesSpace_cluster <- sce$spatial.cluster
```

运行结果解读：`clusterPlot` 画出一张组织切片，不同颜色代表不同空间域。与第 08 章的 `SpatialDimPlot` 对比，你会看到两个明显区别：一是分块更“整”——颜色区域边界平滑、成片分布，因为 BayesSpace 强制相邻 spot 同域；二是分块往往对应组织学的解剖结构。`qTune`

的输出是每个 q 的一组分域结果，肉眼扫一遍就能发现：q 太小会把明显不同的区域硬并在一起，q 太大则出现碎斑。**注意：spatialCluster 基于 MCMC 采样，每次运行结果有细微随机性，正式分析建议设随机种子（`set.seed(123)`）保证可复现。**

SpaGCN 与 STAGATE 一句话对比

BayesSpace 只用表达 + 坐标；另外两个流行工具思路不同，按场景选：

- **SpaGCN**：把空间坐标、表达和（可选的）组织学图像颜色融合成图卷积网络的输入，再聚类。**如果手头有高质量 H&E 图像，SpaGCN 能利用染色信息，通常加分**——但参数更多、更吃调参。
- **STAGATE**：基于图注意力自编码器（graph attention autoencoder），把 spot 邻居关系编码进嵌入，再聚类。对组织边界敏感，但同样依赖超参数。

本书选 BayesSpace 的理由：原理直观（新手容易讲清楚）、参数少（主要就 q 和 d）、结果稳定，是教学与入门的最优解。真实课题若追求极致分域，可以三家都跑再取一致区。

10.3 第二步：邻域富集验证（Squidpy 实操）

分割之后要回答一个尖锐的问题：**这些空间域是真的“细胞扎堆”的结果，还是聚类算法的巧合？**邻域富集分析（neighborhood enrichment）就是干这个的。它的逻辑是：把每个 spot 的邻居（比如六边形网格上相邻的 6 个 spot）统计出来，计算“细胞类型 A 的邻居里出现类型 B”的频率，再与随机置换（permutation）下的期望频率比较，得到 z 分数。z 分数显著为正 = A 与 B 倾向共定位（富集）；显著为负 = 倾向互斥。

```
import squidpy as sq

# 第一步：构建空间邻居图（Visium 是六边形网格，coord_type="generic"，取 6 邻域）
sq.gr.spatial_neighbors(adata, coord_type="generic", n_neighs=6)

# 第二步：邻域富集检验（基于置换检验，cluster_key 是细胞类型标签列）
sq.gr.nhood_enrichment(adata, cluster_key="celltype")

# 第三步：画热图——z 分数，红 = 共定位富集，蓝 = 互斥
sq.pl.nhood_enrichment(adata, cluster_key="celltype")
```

运行结果解读：热图的行列都是细胞类型，格子的 z 分数表示两类细胞的共定位程度（图 10-2 就是它的成品，见 10.5 节）。典型模式：肿瘤细胞 × 肿瘤细胞对角线上深红——肿瘤细胞确实彼此紧挨成片；肿瘤细胞 × CD8 T 细胞显著为正——免疫细胞浸润到肿瘤巢；肿瘤 × B 细胞接近 0 或为负——它们通常不直接相邻。**这套“谁和谁挨着”的证据，正是第 11 章通讯分析的前置证据：配体-受体互作要有意义，前提是两类细胞真的空间相邻。**

co-occurrence 补充：除了“是不是邻居”，有时你还想知道“随着距离增加，共现如何衰减”。co-occurrence 分析给出随距离变化的共现概率曲线，能区分“紧密接触”与“松散共处”：

```
# 共现分析：随距离变化的共现概率（补充 nhood_enrichment 的视角）
sq.gr.co_occurrence(adata, cluster_key="celltype")
sq.pl.co_occurrence(adata, cluster_key="celltype")
```

运行结果解读：输出一组曲线，横轴是距离（spot 数），纵轴是共现概率。如果肿瘤与巨噬细胞的曲线在近距离处明显高于随机线，说明二者紧密相邻——这个信息在后面的通讯重编程故事里会反复用到。

10.4 第三步：生态位命名与特征化

命名规则

分割出的空间域（BayesSpace cluster）没有自带名字，需要我们把“细胞组成 + 位置 + 先验知识”综合起来命名。本书采用的教学命名规则如下（与 00-TECH 速查第 4 节一致；**注意：名称仅为教学约定，真实课题应基于数据 + 先验命名并给出依据**）：

生态位	典型组成	特征通路 / marker
肿瘤核心（tumor core）	肿瘤细胞为主、血管内皮	EPCAM/KRT、VEGF 高
侵袭前沿（invasive front）	肿瘤 + CAF 交界、EMT 特征	VIM/ZEB1/TGF-β 高
免疫浸润区（immune-infiltrated region）	T/B/巨噬细胞富集	CD3D/CD8A/MS4A1 高
基质区（stromal region）	CAF、ECM 富集	COL1A1/DCN/PDGFRA 高

命名的实操流程：先按第 09 章的细胞组成给每个空间域画堆叠条形图（每个域 = 一根柱子，看各细胞类型占比），再对照上表“对号入座”。比如一个域的巨噬细胞和 T 细胞占比最高，就命名为免疫浸润区；一个域肿瘤细胞占比 90% 且位于切片中央，就是肿瘤核心。命名要**可证伪**：给出你依据的 marker 证据，而不是凭感觉。

特征化：marker 基因与通路富集

命名之后，用两个分析把每个生态位的“个性”量化出来：一是差异表达基因（每个生态位相对其他生态位高表达的基因），二是通路富集（这些基因富集在哪些生物学通路）。

```
# 把生态位作为分组变量，找每个生态位的特征基因（BH-FDR 校正，FindAllMarkers 内置）
Idents(obj) <- obj$niche   # niche 列 = 命名后的生态位标签
niche_markers <- FindAllMarkers(obj, only.pos = TRUE,
                                min.pct = 0.25, logfc.threshold = 0.5)
# 查看侵袭前沿的 top 特征基因
head(niche_markers[niche_markers$cluster == "侵袭前沿", ], 10)
```

```
# Hallmark 通路富集（ORA，过表达分析）
# 先获取 Hallmark 基因集（msigdbr 包：BiocManager::install("msigdbr")）
library(msigdbr); library(clusterProfiler)
hallmark <- msigdbr(species = "Homo sapiens", category = "H") %>%
  dplyr::select(gs_name, gene_symbol)

# 对"侵袭前沿"的特征基因做富集（其他生态位同理，循环即可）
res <- enricher(gene = niche_markers$gene[niche_markers$cluster == "侵袭前沿"],
                TERM2GENE = hallmark, pvalueCutoff = 0.05)
head(res@result[, c("ID", "p.adjust", "GeneRatio")])
```

运行结果解读：`FindAllMarkers` 输出每个生态位的特征基因表；`enricher` 输出富集到的 Hallmark 通路与校正后 p 值。教学示例里你常会看到：侵袭前沿富集到 EMT（上皮间质转化）、TGF-β 信号；免疫浸润区富集到干扰素 γ 响应、炎症反应；肿瘤核心富集到血管生成（VEGF 通路）。**注意：通路富集结论要注明统计方法（这里是超几何检验 + BH-FDR 校正），并且"富集到 EMT 通路" ≠ "细胞正在发生 EMT"，前者是计算证据，后者需要实验验证**（第 00-SPEC 第 8 节）。

10.5 结果解读：生态位图与邻域富集热图

图 10-1 是生态位识别的最终结果（模拟数据演示）：

怎么读这张图：左图是成品的"生态位地图"——四种颜色对应四个生态位，边界清晰、成片分布。检查它是否合理，看三点：①肿瘤核心是否位于切片中央的大块区域；②侵袭前沿是否夹在肿瘤核心与基质区之间（"前沿"的字面意思）；③免疫浸润区是否零散分布但成簇。如果某生态位碎成几十个孤立小点，说明分割过细或数据质量有问题，回 10.2 节调 q。右图给出各生态位的 spot 数，方便评估样本量——spot 数太少的生态位（比如 < 50）后面做分区通讯时统计功效不足。

图 10-2 是邻域富集验证结果（模拟数据演示）：

怎么读这张图：对角线格子（A×A）深红说明该类细胞"自己挨自己"，是成片分布的标志；非对角线格子红色说明两类细胞倾向相邻。把图 10-2 与图 10-1 对照：图 10-1 里侵袭前沿的组成是"肿瘤 + CAF 交界"，那么图 10-2 里肿瘤细胞 × CAF 就应该显著富集——两张图互相

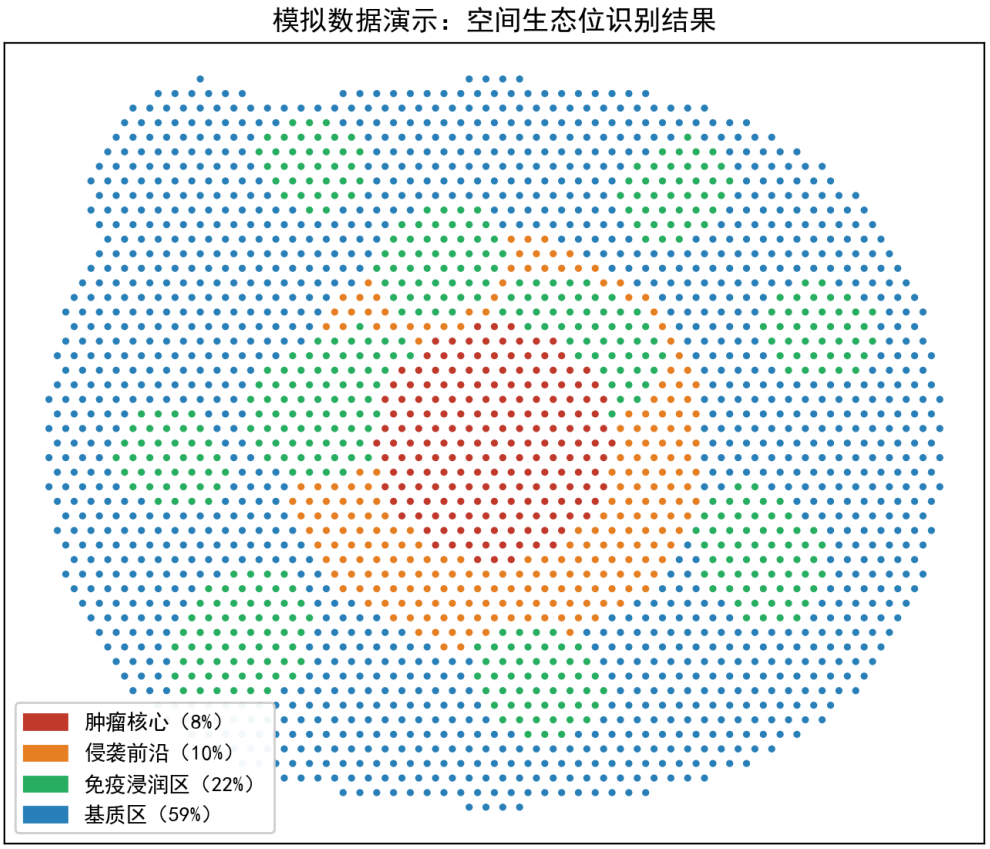

图 15: 图 10-1 空间生态位识别最终结果（模拟数据演示）。左：组织切片按四个生态位着色（肿瘤核心、侵袭前沿、免疫浸润区、基质区）；右：各生态位占比图例与 spot 数。

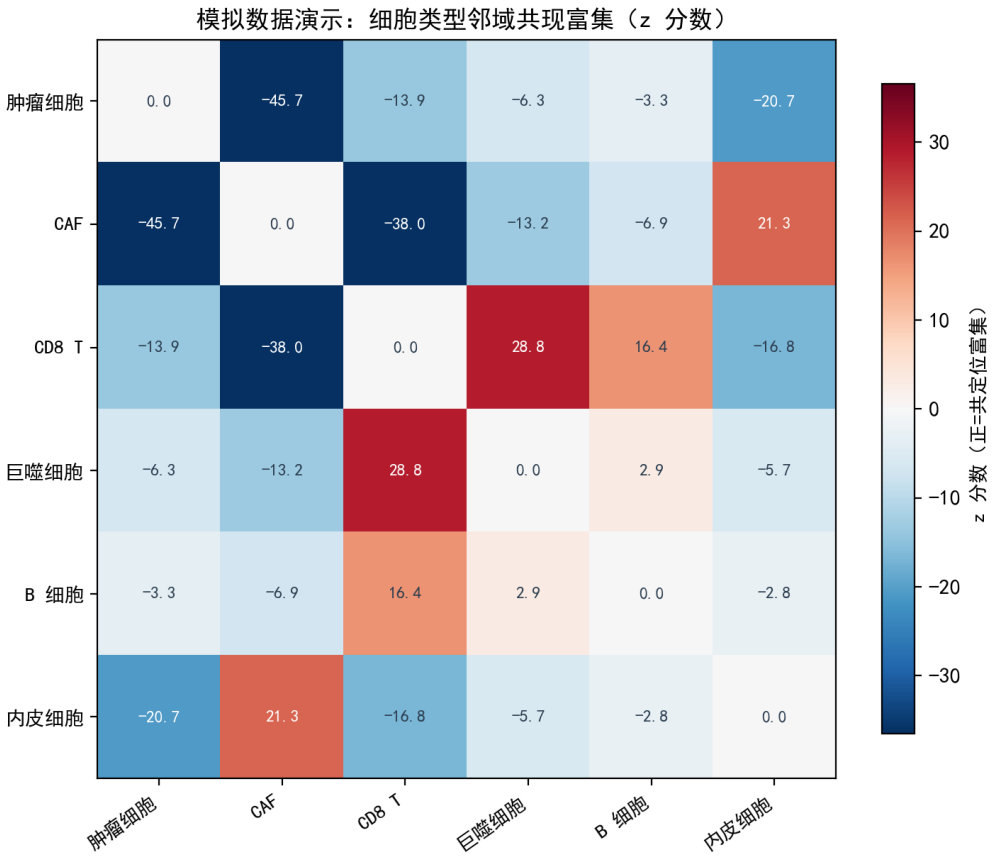

图 16: 图 10-2 细胞类型邻域共现富集热图（模拟数据演示，Squidpy nhood_enrichment 风格）。行列为细胞类型，颜色为 z 分数：红色表示两类细胞显著共定位（互为邻居的频率高于随机），蓝色表示显著互斥。

印证，生态位的划分才站得住脚。**注意：本图为模拟数据演示，仅用于展示“真实分析会长什么样”。**

10.6 生态位数量的生物学判断

q 选多少？统计上可以看模型拟合，但更重要的判断标准是**生物学可解释性**。参考原则：

- **4–7 个较常见**：太少（< 4）会把生物学上不同的区域并到一起，后面“重编程”比较无从谈起；太多（> 10）则出现无法解释的碎片。
- **宁少勿多**：每个生态位都要能回答“它是什么、为什么存在”。一个说不清的生态位是分析的负担而不是收获。
- **三个硬性检查**：①每个生态位有独特、可解释的细胞组成（堆叠条形图差异明显）；②空间上连成片（不是碎点）；③能在 H&E 切片上找到对应的组织学结构（肿瘤巢、纤维间质、淋巴浸润灶）。三条都过，这个 q 才敢用。

实践技巧：把 q 从 4 跑到 8，画出每个 q 的生态位地图和组成条形图，放在一起看“哪个 q 让每个域都说得通”——这就是 10.7 节练习要做的事。

10.7 动手练习：跑 BayesSpace，比较 q = 4 / 6 / 8

```
library(patchwork)                    # 用 + 拼接多张 ggplot 图
set.seed(123)                         # 保证 MCMC 可复现
sce <- as.SingleCellExperiment(obj)
sce <- spatialPreprocess(sce, platform = "Visium", n.PCs = 15)

# 对三个 q 分别聚类，结果存进不同列
sce$niche_q4 <- spatialCluster(sce, q = 4, platform = "Visium",
                               d = 15)$spatial.cluster
sce$niche_q6 <- spatialCluster(sce, q = 6, platform = "Visium",
                               d = 15)$spatial.cluster
sce$niche_q8 <- spatialCluster(sce, q = 8, platform = "Visium",
                               d = 15)$spatial.cluster

# 三个结果并排画在组织切片上
clusterPlot(sce, label = "niche_q4") + clusterPlot(sce, label = "niche_q6") +
  clusterPlot(sce, label = "niche_q8")
```

运行结果解读：三张切片图并排，你能直观看到 q 增大时生态位如何“分裂”：$q = 4$ 时基质区可能只有一大片；$q = 8$ 时同一个区域被切成两三块。接着按 10.4 节的流程给每套结果命名、画细胞组成条形图，回答：**$q = 6$ 相比 $q = 4$ 多分出来的域，有独特的细胞组成吗？$q = 8$**

相比 q = 6 多出来的域，能解释吗？把回答写进你的分析笔记——这就是你为最终 q 值准备的“决策记录”，论文方法部分可以直接引用。

本章小结

本章完成了课题的核心一步：用三步法把组织切成有生物学意义的生态位。第一步用 BayesSpace（马尔可夫随机场空间先验）做空间域分割并选定 q；第二步用 Squidpy 邻域富集验证细胞共定位不是巧合；第三步用 marker 与 Hallmark 通路给每个生态位命名和画像。至此你手里有了“生态位 × 细胞类型”的完整空间地图——下一章开始，我们将在这张地图上做细胞通讯分析，回答本课题真正的问题：不同生态位的通讯网络有什么不同，谁在“重编程”谁。

练习与思考

1. **概念题**：BayesSpace 与第 08 章 Seurat 聚类（FindClusters）的本质区别是什么？（提示：从“是否使用空间位置”和“如何保证相邻 spot 同域”两个角度回答）
2. **概念题**：为什么说“邻域富集验证”是生态位识别中不可省略的一步？如果跳过它，分割结果可能有什么问题？
3. **动手题**：按 10.7 节跑 q = 4/6/8 三套结果，用细胞组成堆叠条形图比较，说明你最终选择哪个 q 以及理由。
4. **动手题**：用 `SpatialFeaturePlot` 把侵袭前沿的两个特征基因（如 VIM、ZEB1）画到切片上，验证它们是否真的在“侵袭前沿”生态位高表达。
5. **思考题**：你的数据里免疫浸润区被分成了两个生态位，一个 T 细胞多、一个巨噬细胞多。你会把它们合并成一个“免疫浸润区”，还是保留为两个？请给出你的判断标准（提示：联系第 11–13 章“分区通讯”对生态位数量的需求）。

第 11 章细胞通讯分析基础：CellChat

> 本章对应脚本：`code/06_cellchat.R`。前置：完成第 8–10 章，手头有一个经过预处理、注释、生态位划分的 Visium 乳腺癌 Seurat 对象 `obj`。

从这一章开始，我们进入全书的核心轨道——细胞通讯（cell-cell communication, CCC）分析。前 10 章回答的是 “肿瘤微环境（tumor microenvironment, TME）里有什么、长在哪里”；从本章起要回答 “它们之间在说什么”。肿瘤不是一堆细胞的简单集合，而是靠细胞通讯组织起来的 “社会”：肿瘤细胞招募血管、驯化免疫细胞，成纤维细胞为肿瘤铺路……这些对话正是肿瘤进展的引擎。本章用 R 包 CellChat v2 作为主工具，讲清楚三件事：为什么需要推断通讯、CellChat 是怎么推断的、跑完流程后怎么读图。

本章目标

1. 理解为什么 “表达≠通讯”，以及配体-受体（ligand-receptor, L-R）证据为什么是必需的
2. 用 “讲人话” 的方式理解 CellChat 原理：L-R 数据库→通讯概率→排列检验→通路聚合
3. 学会安装 CellChat v2，并从 Seurat 对象构建 CellChat 对象（重点：Visium 场景的 “细胞类型” 从哪来）
4. 掌握完整分析流程（subsetData → computeCommunProb →… → aggregateNet）与 5 种经典可视化
5. 学会 “读图说话”，并牢记：通讯推断是计算预测，不是直接测量

1 为什么推断细胞通讯：表达≠通讯

1.1 表达矩阵只是 “原材料清单”

每个 spot（或单细胞）的表达矩阵，本质是一张 “原材料清单”：它告诉我们某个基因在这个位置转录了多少。但 “细胞 A 表达了配体基因 X” 远不等于 “细胞 A 正在跟别人通讯”。想象一场对话：A 说了话（表达配体），但 B 得长着耳朵（表达受体）才听得见；两人还得离得够近（空间邻近），声音才传得到；而且 A 说的可能是 “废话”（蛋白没有功能活性，或存在抑制物、诱饵受体）。所以从 “表达” 到 “通讯”，中间隔着至少四道证据关卡：

1. **受体证据**：接收者细胞群必须表达对应的受体；

2. **空间证据**：配体细胞与受体细胞在空间上足够接近（旁分泌需要邻近，近分泌/接触型需要直接接触）；
3. **功能证据**：互作真的发生且产生下游效应（mRNA 水平只是代理指标，蛋白质、翻译后修饰、可溶/膜结合形式都会影响）；
4. **统计证据**：观察到的共表达不是随机碰巧。

1.2 配体-受体互作数据库：通讯的“字典”

既然表达矩阵不能直接告诉我们“谁在跟谁说话”，我们就需要外部知识——配体-受体（L-R）互作数据库。它相当于一本经过人工整理的“通讯字典”：条目是“配体→受体”（如 CXCL12 → CXCR4、CD274/PD-L1 → PDCD1/PD-1），并注明这条互作属于哪类信号。推断算法的任务，就是拿着表达矩阵去查这本字典：发送者表达配体、接收者表达受体，且统计上显著——就记一条“疑似通讯”。CellChat 用的字典叫 CellChatDB（见 2.1 节）。请注意：查字典只能发现“字典里有的词”，全新或未注释的互作是推断不出来的，这一点在第 7 节还会强调。

2 CellChat 原理（讲人话）

2.1 CellChatDB：三类互作的数据库

CellChatDB 是 CellChat 自带的 L-R 数据库，互作按生物学类别分为三类：

- **分泌信号（secreted signaling）**：可溶性配体分泌到胞外，作用于邻近或远处细胞的受体，如趋化因子、细胞因子、生长因子；
- **ECM-受体（ECM-receptor）**：细胞外基质（如胶原、纤连蛋白）与细胞表面受体（如整合素）的互作，如 FN1-ITGA5、SPP1-CD44；
- **细胞-细胞接触（cell-cell contact）**：膜结合配体与相邻细胞受体的直接接触，包含共刺激与共抑制分子，如 PD-L1/PD-1、CD86-CTLA4。

v2 版本大幅扩展了数据库：支持更多物种（人类、小鼠、斑马鱼等）与更多互作条目，并新增了空间转录组分析模式（见 3.3 节）。

2.2 通讯概率：质量作用定律的思想

CellChat 估计“细胞群 A 通过配体 L 向细胞群 B 通讯”的概率，核心思想来自化学里的质量作用定律（law of mass action）：反应速率与反应物浓度乘积成正比。类比过来：**通讯概率∝发送者群中配体的平均表达 × 接收者群中受体的平均表达**。具体实现时还做了几件事：

- 用三均值（triMean，一种去掉两端极端值后的稳健平均）聚合群内细胞的表达，避免个别高表达细胞带偏整个群；
- 考虑“群里有多少比例的细胞表达该基因”（只有少数细胞表达时通讯不可靠）；

- 多亚基配体/受体（如 TGF-β 需要 TGFBR1 与 TGFBR2 两个亚基）按复合体方式合并；
- 乘以细胞群大小比例（population.size）：大细胞群发出/接收的信号总量更大。

2.3 排列检验：这个通讯显著吗

表达总是有噪声的，A 表达配体、B 表达受体也可能纯属巧合。CellChat 用排列检验（permutation test）判断显著性：把细胞类型标签随机打乱（保持各类数量不变），重新计算通讯概率，重复约 1000 次，得到“纯靠运气”时的概率分布；把真实观测值放进去，如果它大于 95% 的随机值，就认为显著（$p < 0.05$）。这一步由 `computeCommunProb` 内置完成，不需要我们手动实现。

2.4 通路级聚合：从基因对到通路网络

单条 L-R 对太细碎（人类数据库里有几千对），不利于讲故事。CellChat 提供两级聚合：

- `computeCommunProbPathway`：把属于同一信号通路的多条 L-R 对合并（如 TGFb 通路包含 TGFB1-TGFBR1、TGFB1-TGFBR2 等多对），得到“通路级”通讯概率；
- `aggregateNet`：再把所有通路汇总成一张全网络，得到每个细胞群对之间的显著互作数量（count）与通讯强度（weight）。

所以 CellChat 输出的是一套分层的网络：L-R 对级→通路级→网络级，我们按需取用。

3 安装 CellChat v2 与对象构建

3.1 安装

CellChat v2 不在 CRAN，需要从 GitHub 安装（[Jin 2024]；仓库 https://github.com/jinworks/CellChat）。依赖包较多（NMF、ComplexHeatmap 等），Windows 用户请先装好 Rtools（Bioconductor 包必需）。安装只需一次：

```
# Windows 用户先装 Rtools (https://cran.r-project.org/bin/windows/Rtools/) 并配置好 PATH
install.packages("devtools")
# 安装依赖与 CellChat v2 (编译需几分钟，耐心等待)
devtools::install_github("jinworks/CellChat")
library(CellChat)
packageVersion("CellChat")   # 本书以 v2.x 为例
```

运行结果解读：若最后一行打印出版本号（如 2.1.x）且没有报错，说明安装成功。常见报错多为缺少系统编译工具（Rtools）或某个依赖包版本过旧，按提示补装即可。

3.2 从 Seurat 构建 CellChat 对象

```
library(Seurat)
# obj 是第 8–10 章的结果：Visium 乳腺癌 Seurat 对象
# group.by 指定”细胞类型”列（第 9 章注释/去卷积得到的 celltype 列）
DefaultAssay(obj) <- ”SCT”                          # 使用 SCTransform 归一化后的数据
cellchat <- createCellChat(object = obj, group.by = ”celltype”)
# 载入人类配体-受体数据库（CellChatDB.human 是包内自带对象）
cellchat@DB <- CellChatDB.human
```

运行结果解读：createCellChat 会打印每个细胞群的细胞数，并提示数据读取完成。这里有一个 Visium 场景的关键点需要理解：**每个 spot 是 1–10 个细胞的混合体**（见 00-TECH 第 1 节），所以“细胞类型”列不可能像单细胞数据那样干净。实践中 celltype 列通常来自两种做法：一是注释策略（第 9 章），给每个 spot 一个“主要细胞类型”标签；二是去卷积（deconvolution，如 cell2location）后，把丰度最高的细胞类型作为标签，或按丰度比例对通讯概率加权。本书教程采用“主要细胞类型”标签，理解其混合信号局限即可（见第 7 节）。

3.3 v2 的两种空间模式

CellChat v2 针对空间转录组提供两种模式：**细胞群模式（cell groups）**——把同一类型的细胞/spot 合成一组，比较“组与组”之间的通讯，这也是本书主要使用的模式；**细胞-spot 模式（cell-spot）**——保留单个 spot 的身份，利用空间坐标推断 spot 与 spot 之间的通讯（适合高分平台）。当传入的对象自带空间坐标时，v2 会自动启用空间模式，无需额外设置。

4 完整分析流程

下面的代码是全书通讯分析的“标准动作”，第 12、13 章会反复使用：

```
# ---- CellChat 完整流程（每步都解释”为什么”）----
# 1）只保留数据库中有互作记录的基因，大幅节省内存与时间
cellchat <- subsetData(cellchat)
# 2）找出每个细胞群中相对其他群”过表达”的基因（判断互作强度的基础）
cellchat <- identifyOverExpressedGenes(cellchat)
# 3）标记”配体和受体都过表达”的 L-R 对，缩小候选范围
cellchat <- identifyOverExpressedInteractions(cellchat)
# 4）核心步骤：质量作用定律估计通讯概率 + 内置排列检验（约 1000 次）
#    可选并行加速：future::plan(”multisession”, workers = 4)
cellchat <- computeCommunProb(cellchat, type = ”triMean”)
# 5）过滤：要求每个细胞群至少有 min.cells 个细胞，小群不参与推断
cellchat <- filterCommunication(cellchat, min.cells = 10)
# 6）通路级聚合：把同一通路的多条 L-R 对合并
```

```
cellchat <- computeCommunProbPathway(cellchat)
# 7) 网络级聚合：得到 net$count（显著互作数量）与 net$weight（通讯强度）
cellchat <- aggregateNet(cellchat)
# 8) 保存结果，方便下次直接读取，不必重跑
saveRDS(cellchat, file = "results/cellchat_global.rds")
```

运行结果解读：第 4 步最耗时（几分钟到几十分钟，取决于细胞群数与基因数）。完成后可用 `cellchat@net$count` 查看“群 × 群”的显著互作数量矩阵，`cellchat@net$weight` 查看通讯强度矩阵，`dim(cellchat@netP$prob)` 查看通路级概率数组（第三维是通路名）。此外，`cellchat@net$pval` 保存了每条 L-R 对的排列检验 p 值，第 13 章做差异分析时会用到。

5 经典可视化逐个讲解

这一节介绍 5 张最常用的“看图工具”，它们分别回答不同层次的问题：netVisual_circle 看“整体网络长什么样”，netVisual_bubble 看“具体哪对配体-受体在说话”，netVisual_heatmap 看“谁发给谁的整体强度”，rankNet 看“哪些通路是主旋律”，netAnalysis_signalingRole 看“每个细胞群在对话中的角色”。建议的学习顺序：先用 circle 建立整体印象，再用 bubble 与 heatmap 追问细节，最后用 rankNet 与 signalingRole 提炼结论。

5.1 netVisual_circle：圆形网络图

```
# 圆形网络：节点 = 细胞群，边 = 显著通讯；边越粗 = 强度越大
netVisual_circle(cellchat,
                 vertex.weight = table(cellchat@idents),
                 weight.scale = TRUE,
                 label.edge = FALSE,
                 title.name = "通讯数量")
```

解读要点：先看“有哪些边”（哪些细胞群在对话）、再看“边有多粗”（对话有多强）、最后看“节点大小”（通常代表细胞数）。

5.2 netVisual_bubble：气泡图

```
# 气泡图：行 = 指定通路的 L-R 对，列 = "发送者→接收者"
# 气泡大小 = 通讯概率，颜色 = p 值（越红越显著）
netVisual_bubble(cellchat, signaling = c("TGFb", "MIF", "CXCL"))
```

气泡图适合回答“具体是哪一对配体-受体在说话、说得多响”。

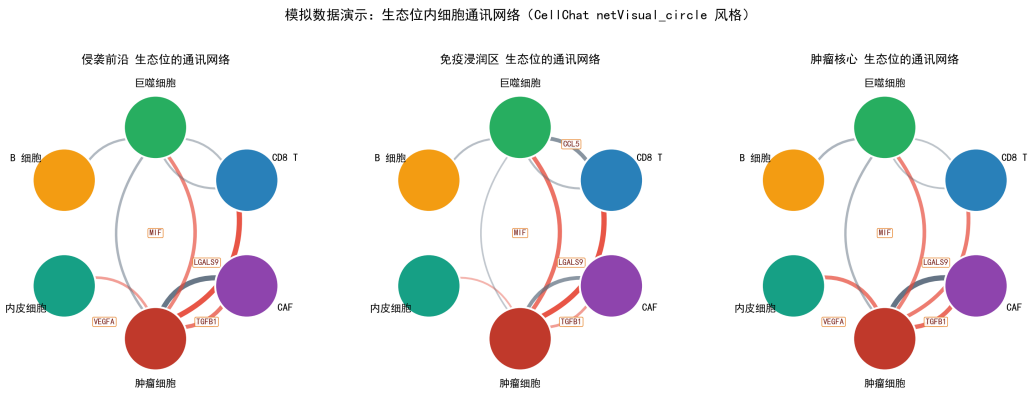

图 17: 图 11-1 三个生态位的通讯网络对比（模拟数据演示）。圆形网络图中每个节点代表一种细胞类型，边代表细胞群之间的显著通讯，边越粗通讯强度越大；图内标注了各生态位最强的配体-受体信号。

5.3 netVisual_heatmap：热图

```
# 热图：发送者（行）× 接收者（列）的通讯强度，适合全局扫一眼
netVisual_heatmap(cellchat, signaling = "TGFb", type = "heatmap")
```

5.4 rankNet：通路排序

```
# 按"相对贡献"给所有通路排序：哪些通路是这台组织的"主旋律"
rankNet(cellchat)
```

5.5 netAnalysis_signalingRole：信号角色

```
# 每个细胞群作为"发送者（outgoing）/接收者（incoming）"的贡献
# 左图：各群的发送/接收总强度条形图；右图：二维散点定位角色
netAnalysis_signalingRole(cellchat)
```

解读：落在右上角的细胞群是“既爱说又爱听”的中心节点，例如肿瘤核心的内皮细胞往往同时接收 VEGF 信号并分泌趋化因子。

6 结果解读示例：读图说话

以图 11-1（模拟数据演示）为例，教大家一套“读图说话”的套路。三张圆形网络并排：侵袭前沿（invasive front）的网络中，肿瘤细胞与 CAF 之间的边最粗，且标注了 TGFb——这说明在肿瘤-基质交界处存在活跃的旁分泌回路；免疫浸润区（immune-infiltrated region）的网络里，

B 细胞与 T 细胞之间的边突出（CXCL 相关），提示存在 T-B 免疫互作；肿瘤核心（tumor core）的网络较小但带有 VEGF 信号，指向血管生成。写进论文时可以这样说："侵袭前沿的通讯网络以肿瘤细胞-CAF 的 TGF-β 回路为特征，免疫浸润区以 T-B 细胞趋化互作为特征，肿瘤核心以血管生成信号为特征，提示不同生态位执行不同的通讯程序。"——注意要加上"这提示……"的措辞，因为一切都还只是计算预测（见第 7 节）。

7 注意事项

1. **通讯推断是计算预测，不是直接测量**。CellChat 的依据是 mRNA 表达，而真正的通讯还受蛋白水平、翻译后修饰、空间距离、抑制因子等影响。结论落地前需要空间共定位证据（Squidpy 邻域富集，见第 10 章）、表达证据，必要时做实验验证（抗体中和、基因敲除、类器官共培养）。
2. **spot 混合信号问题**：Visium 的 spot 覆盖 1–10 个细胞，注释标签本身有误差，通讯推断会继承这个误差。解读时不要把"spot 之间的通讯"说成"细胞之间的通讯"。
3. **单样本是探索性的**：一个切片 = n = 1，个体差异无法评估；任何"上调/下调"结论都需要生物学重复支持。
4. **参数透明**：`type`（聚合方式）、`min.cells`、显著性阈值都会影响结果，写作时要把参数写进方法部分。
5. **数据库的边界**：CellChatDB 收录有限，未收录或新型互作不会被发现，阴性结果要谨慎下结论。

本章小结

本章把通讯推断从"黑盒"变成了"白盒"：表达只是原材料，通讯需要 L-R 字典 + 统计推断；CellChat 用质量作用定律估计通讯概率、用排列检验控制假阳性、用两级聚合把几千对互作浓缩成通路与网络。我们跑通了完整流程，并学会了 5 种经典可视化的读法。下一章将把这套流程"搬进"每个生态位内部——这正是全书创新点的操作化起点。

练习与思考

1. 请列出从"表达"到"通讯"之间缺失的至少 3 个证据环节，并说明 CellChat 只能覆盖其中哪些。
2. 排列检验在 CellChat 中扮演什么角色？如果不做这一步，推断结果会有什么问题？
3.（动手题）用本书示例乳腺癌数据构建 CellChat 对象并跑完第 4 节流程，再用 `netVisual_bubble` 找出免疫浸润区中 T 细胞与 B 细胞之间最强的 5 个 L-R 对。
4.（思考题）如果一个 spot 同时被注释为"肿瘤细胞"和"CAF"（混合信号），它对通讯推断会造成什么偏倚？去卷积方法能改善吗？

第 12 章生态位内细胞通讯分析

本章对应脚本：`code/07_niche_cellchat.R`。前置：第 10 章的生态位列（`niche`）、第 11 章的 CellChat 流程，以及运行完第 11 章后的全局 CellChat 对象。

上一章我们把整张切片当成一个整体，分析了“全局通讯网络”。但肿瘤微环境在空间上远非均质：同一个切片上，侵袭前沿与肿瘤核心的细胞组成、缺氧程度、免疫状态天差地别。如果只看全局网络，这些空间差异会被“平均”掉。本章的核心操作只有一句话：**把数据按生态位拆开，每个生态位独立跑一遍 CellChat，得到“生态位特异的通讯程序”（niche-specific communication program）**。这一步看似简单，却是全书创新点（以生态位视角替代全局视角）的操作化起点。

本章目标

1. 理解“全局通讯平均掉空间异质性”的机制，以及分区分析为什么能发现新信号
2. 学会用 Seurat `subset` 按生态位拆分数据，并把第 11 章流程封装成可复用函数
3. 掌握生态位内通讯特征的三类解读示例（T-B 互作、肿瘤-CAF、血管生成）
4. 会用表格整理各生态位的“主要发送者/接收者/通路”
5. 完成侵袭前沿生态位的完整 CellChat 实战练习

1 核心思路：为什么要按生态位拆开分析

1.1 全局分析如何“平均掉”空间异质性

设想真实的生物学场景：TGF-β 通讯只发生在侵袭前沿的肿瘤细胞与 CAF 交界处，而肿瘤核心和免疫浸润区几乎没有。全局分析会把整张切片的表达混在一起：TGFB1 的平均表达被“稀释”，通讯概率被压低，甚至低于显著性阈值——于是这个真实存在的、空间特异的信号在全局网络里“消失”了。反过来，某种在全局看起来很强的信号，可能只是某个局部区域的贡献，全局图却会让人误以为“处处都强”。一句话：**全局网络是“平均人”，生态位网络才是“当事人”**。用一个数字例子帮助理解：假设侵袭前沿的 TGFB1 平均表达为 8、其余区域的为 1，全局分析会把它们混在一起求平均（约 3），通讯概率随之被压低；分区分析直接用 8 计算前沿的通讯概率，信号几乎不受其他区域稀释。同理，某条通路可能因稀释而跌破显著性阈值，从“显著”变成“不显著”——这正是“平均掉”的含义。

1.2 生态位特异的通讯程序

把数据按生态位拆开后，每个生态位内部的通讯网络反映的是“这个微环境独有的对话模式”，我们称之为生态位特异的通讯程序。图 12-1（模拟数据演示）显示，各生态位的通讯数量与总强度差异明显：侵袭前沿通讯最多最强，免疫浸润区次之，肿瘤核心与基质区规模较小。这种差异本身就是生态位生物学（缺氧、EMT、免疫应答）在通讯层面的投影——也就是说，**生态位不仅细胞组成不同，连“对话方式”都不同**。

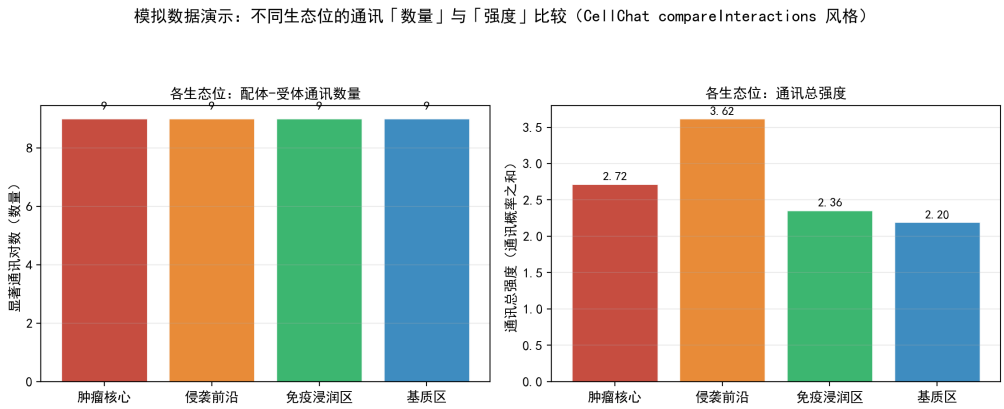

图 18: 图 12-1 各生态位通讯「数量」与「强度」比较（模拟数据演示，compareInteractions 风格）。柱高分别代表显著通讯数量与总通讯强度，可见不同生态位的通讯规模差异明显，侵袭前沿最强。

2 实操：按生态位拆分并独立建 CellChat

2.1 准备：确认生态位列

假设 `obj` 的 meta.data 里有第 10 章生成的 `niche` 列，取值为：肿瘤核心、侵袭前沿、免疫浸润区、基质区。

```
table(obj$niche)   # 看看每个生态位有多少个 spot
```

运行结果解读：如果某个生态位的 spot 数很少（比如少于 100），后面的 `min.cells` 就要相应调小，而且结论要更谨慎——细胞群太小，通讯推断的统计功效不足。若某个生态位的细胞类型数过少（比如只剩 1–2 类），通讯网络会退化成单点互作，此时建议把该生态位与相邻生态位合并后重跑，并在方法部分如实说明。

2.2 把流程封装成函数

第 11 章的 7 步流程在这里完全一样，只是输入对象不同。我们把它封装成函数，避免复制粘贴出错，也为第 13 章批量比较打好基础：

```
# 输入：一个生态位的 Seurat 子集；输出：该生态位的 CellChat 对象
run_niche_cellchat <- function(obj_sub, group.by = "celltype", min.cells = 10) {
  cellchat <- createCellChat(object = obj_sub, group.by = group.by)
  cellchat@DB <- CellChatDB.human
  cellchat <- subsetData(cellchat)
  cellchat <- identifyOverExpressedGenes(cellchat)
  cellchat <- identifyOverExpressedInteractions(cellchat)
  cellchat <- computeCommunProb(cellchat, type = "triMean")
  cellchat <- filterCommunication(cellchat, min.cells = min.cells)
  cellchat <- computeCommunProbPathway(cellchat)
  cellchat <- aggregateNet(cellchat)
  return(cellchat)
}
```

2.3 按生态位循环运行

```
library(Seurat); library(CellChat)
niches <- unique(obj$niche)
cc_list <- list()                              # 用列表装每个生态位的结果
for (n in niches) {
  message("正在分析生态位：", n)
  obj_sub <- subset(obj, subset = niche == n)   # 只保留该生态位的 spot
  cc_list[[n]] <- run_niche_cellchat(obj_sub, min.cells = 10)
}
saveRDS(cc_list, file = "results/cc_list_by_niche.rds")   # 保存，第 13 章继续用
```

运行结果解读：循环会逐个生态位打印分析进度。若某生态位报“细胞类型太少”类错误，把该生态位的 `min.cells` 调低（如 5）重试；若某生态位只剩一两种细胞类型，通讯网络会非常稀疏——这本身就是一个发现（该生态位细胞组成单一），但要如实报告，不要强行解读。

2.4 一个常见的小坑

`subset(obj, subset = niche == n)` 要求 meta.data 里确实存在 `niche` 列。如果列名与 Seurat 内部槽位冲突，`subset` 会报错，建议用 `obj$niche <- ...` 显式赋值，并检查 `colnames(obj@meta.data)` 确认列名无误。

2.5 每个生态位的网络可视化

分区分析的一个额外好处是“图更好讲”：把 4 个生态位的 `netVisual_circle` 分别保存后并排拼接（如图 11-1 的三联图），读者一眼就能看出“不同生态位连的边都不一样”。拼图时注

意统一边的粗细比例尺，否则两张图之间无法比较粗细；同时为每个面板标注生态位名称与最强的 L-R 对，图注里写明“模拟数据演示”。

```
# 每个生态位分别保存网络图，再在文稿中并排拼接
#  (netVisual_circle 输出基础图形，不能用 patchwork 直接组合)
for (n in names(cc_list)) {
  png(paste0("figures/comm_network_", n, ".png"),
      width = 6, height = 6, units = "in", res = 300)
  netVisual_circle(cc_list[[n]], title.name = n)
  dev.off()
}
```

3 生态位内通讯特征解读示例

拿到 4 个生态位的网络后，怎么讲出生物学故事？建议按“三步走”：第一步看网络规模（图 11-1 中哪张图边最多、最粗）；第二步看主要细胞对（谁和谁在对话）；第三步看最强通路（对话的内容是什么）。下面结合文献与本书模拟数据（见图 11-1），给出三个典型模式的解读示例——它们是全书模拟数据的设计蓝本，你在真实数据上看到的不一定完全一样，但解读方法相同：

- **免疫浸润区：T-B 互作**。免疫浸润区富集 T 细胞与 B 细胞，典型信号是趋化因子回路，如 CXCL13-CXCR5（B 细胞向 T 细胞区域趋化）与 CCL19/CCL21-CCR7。这类 T-B 互作与“免疫 hub”（immune hub）概念一致：结直肠癌研究中，B 细胞、滤泡辅助 T 细胞与巨噬细胞在空间上聚集形成 hub，并与良好预后相关 [Pelka 2021]。读图时如果看到 T → B 的 CXCL 边，可以写“提示该生态位存在组织化的免疫应答结构”。
- **侵袭前沿：肿瘤-CAF 互作**。侵袭前沿是肿瘤细胞与基质细胞（尤其 CAF）的交界，EMT 特征显著（VIM/ZEB1/TGF-β 高）。典型信号是 TGFb 通路（TGFB1-TGFBR1/TGFBR2）：CAF 分泌 TGF-β 促进肿瘤细胞 EMT 与侵袭，肿瘤细胞反过来也能激活 CAF。读图时注意“谁发谁收”：若 TGFb 边以 CAF →肿瘤为主，就是“基质驱动”的故事。
- **肿瘤核心：血管生成**。肿瘤核心缺氧，VEGF 通路（VEGFA-KDR/FLT1）活跃：肿瘤细胞分泌 VEGFA，内皮细胞表达受体 KDR。读图时看到“肿瘤细胞→内皮”的 VEGF 边，提示该生态位正在招募血管。

以上解读遵循同一句式：**信号→细胞对→生物学含义→“这提示……”**。记住：网络图只告诉你“谁在跟谁说什么”，生物学含义是读者（你）补上的解释，需要文献与后续验证支撑（第 13 章会给出验证方案）。

4 生态位内 vs 全局通讯：为什么分区能发现新信号

对比维度	全局通讯分析	生态位内通讯分析
分析对象	整张切片所有 spot	单个生态位内的 spot
空间信息	忽略（或仅作辅助）	隐含（生态位本身即空间单元）
对局部信号	被平均稀释，可能低于阈值而丢失	保留局部强度，容易被发现
对全局信号	可能把局部强信号误读为全局普遍	明确信号的 “主场” 在哪
生物学解释	“整个组织在对话”	“这个微环境在对话”
典型风险	空间特异信号消失或错位	跨生态位长程信号被截断

举例：全局分析中 TGF-β 通路可能完全不显著（被稀释），而分区分析发现它在侵袭前沿是排名第一的通路——这就是 “分区分析能看到全局看不到的信号” 的典型场景。当然，分区也有代价：跨生态位的长程信号（如免疫浸润区分泌的细胞因子作用于肿瘤核心）会被截断。另一个典型场景是免疫浸润区的 CXCL 信号：全局分析中趋化因子通讯可能被大片肿瘤区域的噪声淹没，分区分析则能干净地看到 T 细胞→ B 细胞的 CXCL13-CXCR5 边。所以分区分析与全局分析是**互补关系，不是替代关系**：全局网络负责 “有没有”，生态位网络负责 “在哪、多强”。第 13 章的差异分析正是建立在这两层之上。

5 结果整理：各生态位通讯网络汇总表

把 4 个生态位浓缩成一张表，是 “从结果到故事” 的第一步：

```
# 汇总每个生态位的” 主要发送者/主要接收者/主要通路”
for (n in niches) {
  cc <- cc_list[[n]]
  w <- cc@net$weight                          # 群 × 群的通讯强度矩阵
  sender   <- rowSums(w)                      # 行和 = 该群作为发送者的总强度
  receiver <- colSums(w)                      # 列和 = 该群作为接收者的总强度
  top_sender   <- names(sort(sender,   decreasing = TRUE))[1]
  top_receiver <- names(sort(receiver, decreasing = TRUE))[1]
  pw <- apply(cc@netP$prob, 3, sum)           # 每个通路的全网络总概率
  top_pathways <- names(sort(pw, decreasing = TRUE))[1:3]
  cat(” 生态位:”, n, ”\n”,
      ”  主要发送者:”, top_sender, ”\n”,
      ”  主要接收者:”, top_receiver, ”\n”,
      ”  主要通路:”, paste(top_pathways, collapse = ”, ”), ”\n\n”)
}
```

运行结果解读：把打印内容整理成下面的表格（数值为模拟数据演示示例）：

生态位	主要发送者	主要接收者	主要通路
侵袭前沿	肿瘤细胞、CAF	CAF、肿瘤细胞	TGFb、MIF、ECM
免疫浸润区	T 细胞、巨噬细胞	B 细胞、T 细胞	CXCL、CCL、IL10
肿瘤核心	肿瘤细胞	内皮细胞、肿瘤细胞	VEGF、ECM
基质区	CAF	CAF、内皮细胞	ECM、TGFb

这张表就是第 13 章 “通讯重编程” 叙事的骨架：生态位之间的差异已经跃然纸上——同样是 “肿瘤细胞说话”，在侵袭前沿说的是 TGFb，在肿瘤核心说的是 VEGF；同样是 “接收者”，免疫浸润区听的是趋化因子，基质区听的是 ECM 信号。

这张汇总表的每一列都有讲究：主要发送者回答 “谁在主导对话”，主要接收者回答 “谁在被影响”，主要通路回答 “对话内容是什么”。三列组合起来，就能给每个生态位起一个 “人设”：侵袭前沿是 “基质驱动的 TGF-β 放大器”，免疫浸润区是 “T-B 趋化网络”，肿瘤核心是 “血管生成发动机”。给生态位起人设不是卖弄文笔，而是逼迫自己把数据读透——如果人设与生物学先验矛盾，往往意味着注释或参数出了问题。

本章小结

本章完成了从 “全局通讯” 到 “生态位通讯” 的方法学跃迁：把数据按生态位拆开、独立构建 CellChat、封装函数批量运行、用表格汇总每个生态位的通讯特征。核心收获是理解 “平均化陷阱”——全局分析会稀释空间特异信号，而分区分析让每个微环境的通讯程序显形。下一章将正式比较生态位之间的差异，把 “差异” 升级为 “重编程”。另外提醒：本章的通讯规模汇总（数量/强度）只是重编程分析的第一个层面，第 13 章还会叠加通路活性与差异配体-受体对比较——三个层面合起来才是完整的 “通讯指纹”。

练习与思考

1. 用一句话解释：为什么全局通讯分析会 “平均掉” 空间异质性？分区分析能发现什么样的全局分析看不到的信号？
2.（动手题）对本书乳腺癌数据跑侵袭前沿生态位的完整 CellChat 流程（第 2 节代码），找出它的主要发送者、主要接收者与排名前三的通路，并写一段 3 句话的解读。
3.（动手题）把 `min.cells` 从 10 改为 3，重新运行免疫浸润区生态位，比较两次结果的通讯数量差异，思考参数对结论的影响。
4.（思考题）分区分析会截断跨生态位的长程信号。如果某细胞因子由免疫浸润区分泌、作用于肿瘤核心，你会怎么设计分析来捕获它？

第 13 章通讯重编程：生态位间差异分析

> 本章对应脚本：`code/08_differential_comm.R`。前置：第 12 章的 `cc_list`（各生态位的 CellChat 对象列表），以及对应的 scanpy 对象 `adata`。

本章是全书的 “高潮”。前两章我们分别拿到了全局通讯网络和生态位内通讯网络；本章要做的是把它们之间的差异 “讲成故事”——这就是 “通讯重编程”（communication reprogramming）。请注意，“重编程” 在这里不是修辞，而是可以操作化的分析对象：生态位之间在通讯数量、强度与通路活性上的系统性差异。读完本章，你将能把 “侵袭前沿被重编程为免疫抑制枢纽” 这样一句话，变成一张张有数据支撑的图表。

本章目标

1. 掌握 “通讯重编程” 的操作化定义（数量、强度、通路活性三个层面）
2. 学会生态位间差异通讯的比较方法（compareInteractions / 手动汇总）
3. 学会识别差异信号通路（rankNet 对比、通路活性热图、netVisual_diff）
4. 学会 “差异配体-受体对” 分析，并讲清 log2FC 与方向的含义
5. 建立 “三重证据闭环”（空间共定位 + 空间通讯活性 + 表达证据），并用 Sankey 图把重编程故事讲完整

1 什么是 “通讯重编程”：操作化定义

“重编程” 的生物学直觉是：肿瘤在塑造微环境，微环境反过来塑造肿瘤，最终整个组织的通讯格局被系统性改写。要让这个概念可检验，必须把它翻译成可计算的指标。本书的操作化定义是：**通讯重编程 = 生态位之间通讯网络在数量（count）、强度（weight）、通路活性（pathway activity）上的系统性差异**。

- **数量**：显著 L-R 互作的条数，衡量网络 “规模”；
- **强度**：通讯概率之和，衡量网络 “音量”；
- **通路活性**：每个信号通路的总通讯概率，衡量网络 “曲目”。

三者合起来，一个生态位的 “通讯指纹” 就完整了。图 12-1（见第 12 章）展示的正是第一个层面：不同生态位的通讯数量与强度差异显著。注意，重编程是**系统级差异**，与 “某一条 L-R 对在不同生态位强弱不同” 这种单点差异不同——前者是格局，后者是细节；本章先看格局（第

2、3 节），再看细节（第 4 节）。这套定义与第 5 章登记的研究假设一一对应：H1（不同生态位的通讯网络存在系统性差异）对应数量与强度层面，H2（侵袭前沿存在免疫抑制性通讯重编程）对应通路活性层面，H3（分区分析能揭示更多空间异质性信号）则贯穿第 12 章与本章的比较逻辑——本章的每一张图，都是在为某条假设提供证据。

2 差异通讯数量/强度比较

2.1 compareInteractions：官方比较函数

当两个生态位（或两个样本）的细胞类型集合完全一致时，可以先把对象合并，再用官方函数比较：

```
# 前提：两个对象的细胞类型集合完全相同（如” 样本 A vs 样本 B 的侵袭前沿”）
# 若集合不同（本书生态位常有此情况），请直接跳到 2.2 手动汇总
cc_merged <- mergeCellChat(cc_list[c(” 侵袭前沿”, ” 肿瘤核心”)],
                           add.names = c(” 侵袭前沿”, ” 肿瘤核心”))
# 比较总通讯数量（count）与总强度（weight）；group.1/group.2 填合并时的条件名
compareInteractions(cc_merged, group.1 = ” 侵袭前沿”, group.2 = ” 肿瘤核心”,
                    measure = c(”count”, ”weight”))
```

运行结果解读：输出两张并排柱状图（正是图 12-1 的风格来源），直观显示 “前沿 vs 核心” 的总通讯数量与总强度差异。如果报 “细胞群数量不一致” 错误，说明两个对象细胞类型集合不同——这正是 2.2 节手动汇总的用武之地。

2.2 手动汇总：更通用的做法

从第 12 章的 `cc_list` 出发，逐生态位统计通讯数量与总强度，一步到位：

```
niche_summary <- data.frame(
  niche          = names(cc_list),
  n_interactions = sapply(cc_list, function(cc) sum(cc@net$count)),  # 显著互作总条数
  total_strength = sapply(cc_list, function(cc) sum(cc@net$weight)), # 通讯概率之和
  row.names = NULL
)
print(niche_summary)
# 转成长表后画柱状图（对应图 12-1 的” 数量” 面板）
library(tidyr); library(ggplot2)
niche_long <- pivot_longer(niche_summary,
                           cols = c(n_interactions, total_strength),
                           names_to = ”measure”, values_to = ”value”)
ggplot(niche_long, aes(x = niche, y = value, fill = measure)) +
  geom_col(position = ”dodge”) + theme_minimal() +
```

```
  labs(x = "生态位", y = "数值", title = "各生态位通讯数量与强度（模拟数据演示）")
```

运行结果解读：通常看到侵袭前沿的 `n_interactions` 与 `total_strength` 最高。注意这只是描述性比较；要下“显著更高”的结论，需要生物学重复与统计检验（见第 8 节注意事项）。

3 差异信号通路识别

3.1 rankNet 对比

`rankNet` 给每个生态位输出“通路贡献排名”。把 4 个生态位的排名并列，就能看出“曲目”差异：

```
# 逐个生态位看通路排名（对比时注意各自排名前列的通路）
for (n in names(cc_list)) {
  cat("==== 生态位：", n, " ====\n")
  print(rankNet(cc_list[[n]]))   # 返回排序后的通路与相对贡献
}
```

运行结果解读：如果侵袭前沿的榜首是 TGFb/MIF，免疫浸润区是 CXCL/CCL，肿瘤核心是 VEGF——“曲目”差异一目了然，这就是重编程的故事素材。

3.2 通路活性热图：生态位 × 通路

把每个生态位的通路总概率拼成矩阵画热图，一张图看完全部差异（对应图 13-2，模拟数据演示）：

```
# 从每个生态位取"通路级总概率"，拼成生态位 × 通路矩阵
pw_list <- lapply(cc_list, function(cc) apply(cc@netP$prob, 3, sum))
all_pw  <- unique(unlist(lapply(pw_list, names)))
pathway_mat <- sapply(pw_list, function(v) v[all_pw])
pathway_mat[is.na(pathway_mat)] <- 0
pathway_mat <- t(pathway_mat)                          # 行 = 生态位，列 = 通路
pathway_mat <- pathway_mat[, colSums(pathway_mat) > 0]     # 去掉全零通路
pheatmap::pheatmap(pathway_mat, cluster_cols = FALSE,
                   main = "生态位 × 通路活性（模拟数据演示）")
```

运行结果解读：热图里能直接看到“前沿高活性”的免疫抑制与侵袭相关通路（TGFb、PD-L1、MIF、GALECTIN），与图 13-2 的风格一致。若提示没有 `pheatmap` 包，先运行 `install.packages("pheatmap")`。

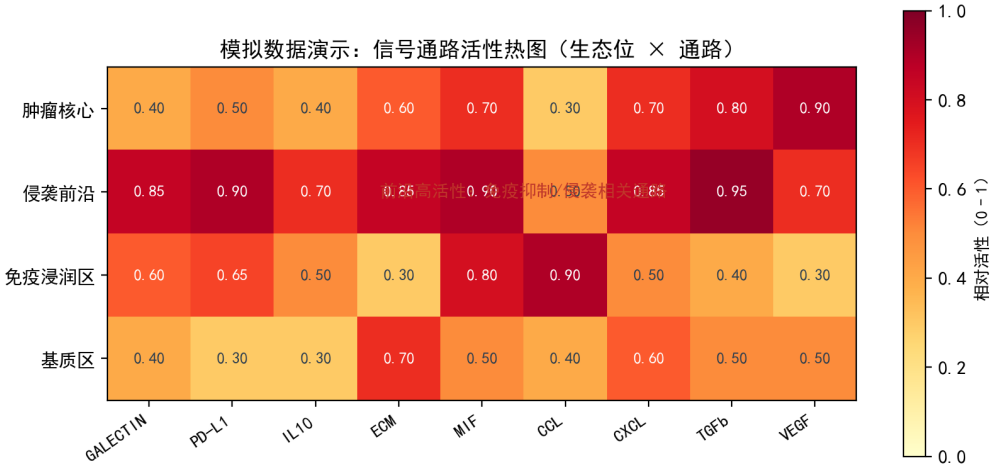

图 19: 图 13-2 生态位 × 信号通路活性热图（模拟数据演示）。行是通路（TGFb、PD-L1、MIF、GALECTIN、VEGF、CXCL、ECM、IL10、CCL），列是生态位；可见侵袭前沿在免疫抑制与侵袭相关通路上活性最高。

3.3 netVisual_diff：网络级差异

如果走 2.1 的合并路线，还能在网络上直接看差异：

```
# 在合并对象上可视化两个生态位的网络差异（红色 = 前沿更强，蓝色 = 核心更强）
netVisual_diff(cc_merged, weight.edge = TRUE, measure = "weight")
```

4 差异配体-受体对分析：谁在具体地 “多说/少说”

通路是 “曲目”，L-R 对才是 “乐句”。跨生态位比较每条 L-R 对的通讯概率，并计算 log2FC：

```
library(ggplot2)
# 提取每个生态位" 每条 L-R 对的全网络平均通讯概率"
pair_prob <- function(cc) apply(cc@net$prob, 3, mean)   # 3D 数组按第三维（L-R 对）求平均
prob_front <- pair_prob(cc_list[[" 侵袭前沿"]])
prob_core  <- pair_prob(cc_list[[" 肿瘤核心"]])
pairs <- union(names(prob_front), names(prob_core))
df <- data.frame(
  pair  = pairs,
  front = prob_front[pairs],
  core  = prob_core[pairs]
)
df[is.na(df)] <- 0
# log2FC > 0: 前沿更强（上调）; log2FC < 0: 核心更强（前沿下调）
df$log2FC <- log2((df$front + 1e-4) / (df$core + 1e-4))   # +1e-4 防止除零
df$direction <- ifelse(df$log2FC > 0.5, " 前沿上调",
                ifelse(df$log2FC < -0.5, " 前沿下调", " 无显著差异"))
head(df[order(-abs(df$log2FC)), ], 10)
```

```r
# 画成点图（对应图 13-1 的左面板风格）
top15 <- head(df[order(-abs(df$log2FC)), ], 15)
ggplot(top15, aes(x = log2FC, y = reorder(pair, log2FC), color = direction)) +
  geom_point(size = 3) + geom_vline(xintercept = 0, linetype = 2) +
  scale_color_manual(values = c("前沿上调" = "#C44E52", "前沿下调" = "#4C72B0",
                                "无显著差异" = "grey50")) +
  theme_minimal() + labs(x = "log2FC（侵袭前沿 / 肿瘤核心）", y = NULL,
                         title = "生态位间差异配体-受体对（模拟数据演示）")
```

log2FC 与方向的含义（务必讲清楚）：log2FC 是对数尺度的比值——log2FC = 1 表示前沿的平均通讯概率是核心的 2 倍，log2FC = -1 表示只有一半。这里的“方向”指的是“谁更强”：红色（前沿上调）表示该 L-R 对在侵袭前沿的通讯概率系统性高于肿瘤核心，例如 PD-L1（CD274-PDCD1）、MIF-CD74、TGFB1-TGFBR2；蓝色（前沿下调）则相反，例如只活跃于核心的血管生成相关对。图 13-1 就是这种分析的标准呈现：左面板气泡图对比两生态位的通讯概率（实心 = 前沿、空心 = 核心，红色 = 前沿上调、蓝色 = 前沿下调），右面板是 Δ 通讯概率条形热图。

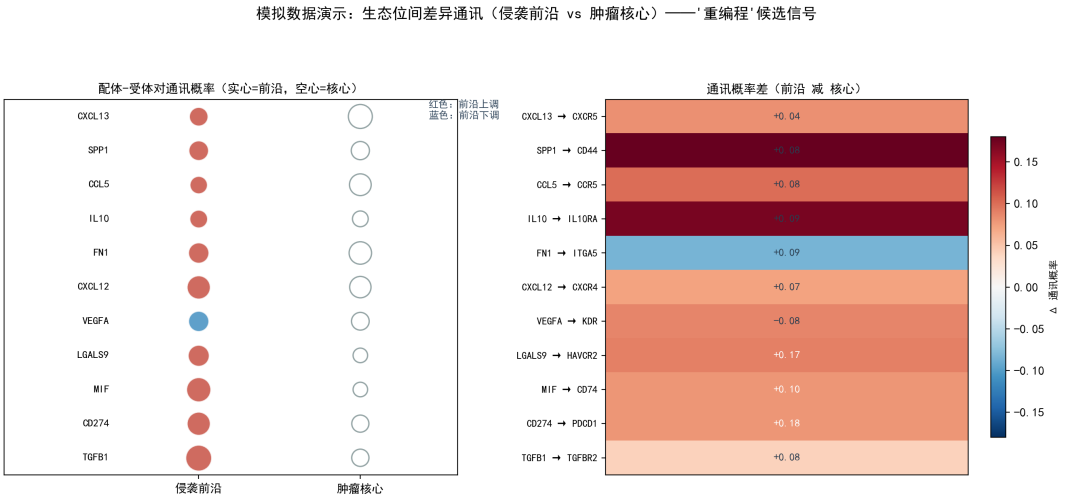

图 20: 图 13-1 侵袭前沿与肿瘤核心的差异通讯（模拟数据演示）。左：气泡图，实心圆代表侵袭前沿、空心圆代表肿瘤核心，红色表示前沿上调、蓝色表示前沿下调；右：Δ 通讯概率条形热图。

注意：这里的 log2FC 是“均值之比”的描述性指标，本身没有 p 值。严谨做法是把 CellChat 给出的每条 L-R 对的排列检验 p 值（存在 `cc@net$pval`）纳入筛选（比如只保留至少在一个生态位显著的对），再结合重复样本做正式统计检验（见第 8 节）。

5 空间活性验证：三重证据闭环

CellChat 的推断本质是“表达层面的共现”，它不知道配体细胞与受体细胞是否真的相邻。本章的可信度建立在三重证据闭环上（对应课题设计中的“空间共定位证据增强”）：

- **证据 1（空间共定位）**：Squidpy 邻域富集，检验“配体细胞类型与受体细胞类型在空间

上相邻”；

- **证据 2（空间通讯活性）**：COMMOT 把空间距离纳入通讯推断，输出每个 spot 的通讯活性；
- **证据 3（表达证据）**：在目标生态位内直接检查配体/受体的表达。

```
# ============ 第 13 章 Python 代码：空间活性验证 ============
import scanpy as sc
import squidpy as sq
import commot as ct

# 前置：adata 是 scanpy 对象（第 9–10 章结果），
# adata.obs["celltype"] 为细胞类型，adata.obs["niche"] 为生态位

# ---- 证据 1：Squidpy 邻域富集（配体细胞与受体细胞是否相邻）----
sq.gr.spatial_neighbors(adata, coord_type="generic", n_neighs=6)
sq.gr.nhood_enrichment(adata, cluster_key="celltype")
z = adata.uns["celltype_nhood_enrichment"]["zscore"]   # z > 0 = 倾向共定位
print(z.loc["CAF", " 肿瘤细胞"])    # 把 CAF/肿瘤细胞换成你数据里的细胞类型名
sq.pl.nhood_enrichment(adata, cluster_key="celltype")  # 全矩阵热图（风格参见第 10 章）

# ---- 证据 2：COMMOT 空间通讯活性（仓库 https://github.com/zcang/COMMOT）----
# dis_thr：通讯距离阈值（μ m）；pathway_sum：聚合到通路级
ct.tl.spatial_communication(adata, database_name="CellChat", dis_thr=250,
                            heteromeric=True, pathway_sum=True)
# 绘制指定通路的空间活性（不同 COMMOT 版本绘图 API 略有差异，以官方文档为准）
ct.pl.plot_cell_communication(adata, cell_type_key="celltype",
                              signaling_type="pathway", communication_key="TGFb")

# ---- 证据 3：表达证据（只看向侵袭前沿生态位）----
sc.pl.dotplot(adata[adata.obs["niche"] == " 侵袭前沿"],
              var_names={" 配体": ["TGFB1", "MIF", "CD274"],
                         " 受体": ["TGFBR2", "CD74", "PDCD1"]},
              groupby="celltype")
```

运行结果解读：三重证据的判读规则是“相互印证”——CellChat 说“前沿有 TGFb 通讯”，Squidpy 说“CAF 与肿瘤细胞确实相邻”，COMMOT 说“TGFb 活性在空间上确实落在前沿”，dotplot 说“配体与受体确实在前沿表达”。四者一致，结论才站得住；任何一环缺失（比如共定位不显著），都要在论文里如实说明，或把该信号降级为“候选”。黄金标准仍是实验验证（中和抗体、基因敲除），计算证据只能提高置信度。

6 重编程故事化：Sankey 流量图

差异分析会产出几十上百条信号，直接堆在论文里没人读得懂。Sankey 图（图 13-3，模拟数据演示）把“来源生态位→重编程通路→靶细胞类型”串成一条河流，带宽∝通讯活性，让“谁通过哪条通路影响了谁”一目了然：

```
# ============ Sankey 流量图（plotly；数据来自第 4–5 节汇总表）============
import plotly.graph_objects as go

# 手工整理三元组：来源生态位 → 通路 → 靶细胞（数值 = 归一化通讯活性，模拟数据演示）
links = [
    (” 侵袭前沿”,   ”TGFb”,  0.35), (” 侵袭前沿”,   ”MIF”,   0.28), (” 侵袭前沿”,   ”PD-L1”, 0.2),
    (” 免疫浸润区”, ”CXCL”,  0.30), (” 免疫浸润区”, ”CCL”,   0.25),
    (” 肿瘤核心”,   ”VEGF”,  0.40),
]
nodes = sorted({s for s, _, _ in links} | {t for _, t, _ in links})
idx   = {n: i for i, n in enumerate(nodes)}
fig = go.Figure(go.Sankey(
    node=dict(label=nodes, pad=15, thickness=20),
    link=dict(source=[idx[s] for s, _, _ in links],
              target=[idx[t] for _, t, _ in links],
              value=[v for _, _, v in links])))
fig.update_layout(title=” 通讯重编程信号流（模拟数据演示）”)
fig.show()
```

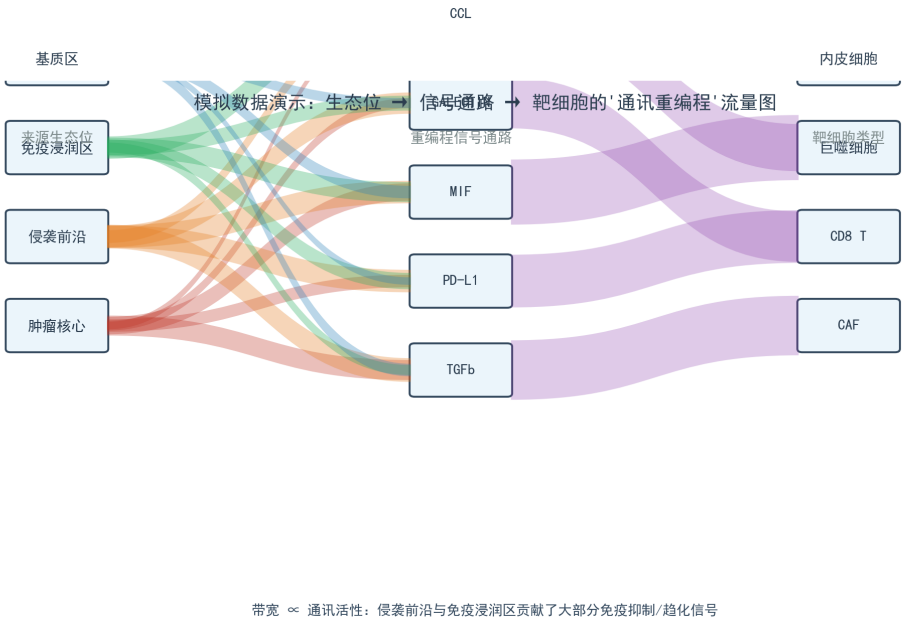

图 21: 图 13-3 通讯重编程信号流 Sankey 图（模拟数据演示）。左列是来源生态位，中列是重编程信号通路，右列是靶细胞类型；带宽与通讯活性成正比，直观呈现“哪个生态位通过哪条通路影响哪些靶细胞”。

解读要领：Sankey 的 “流向” 必须严格忠于数据——来源生态位是 “发送者细胞所在生态位”，靶细胞是 “接收者细胞类型”，中间是通路。不要为了画面好看而添加数据里不存在的连线；每条连线的数值应能在第 4、5 节的结果里找到出处。本章产出的这些 “素材图”（图 13-1 至图 13-3）到第 14 章会按论文 Figure 的版式重新编排（版式示例见 fig17_paper_figure）。

7 结果解读示例：一段示范性 Results 文字

下面是一段 “结果-解读” 示范（数值对应本书模拟数据演示）：

> 与肿瘤核心相比，侵袭前沿生态位的显著通讯数量与总强度均更高（图 12-1）。通路活性分析显示，侵袭前沿在 TGFb、PD-L1、MIF 与 GALECTIN 通路上活性最高，而肿瘤核心以 VEGF 通路为主（图 13-2）。配体-受体对水平上，TGFB1-TGFBR2、CD274-PDCD1、MIF-CD74 在侵袭前沿显著上调（log2FC $>$ 1；图 13-1），且配体细胞（CAF、肿瘤细胞）与受体细胞（肿瘤细胞、免疫细胞）在空间上共定位（Squidpy 邻域富集 z $>$ 2），COMMOT 显示 TGFb 通讯活性集中于侵袭前沿。这些结果提示，侵袭前沿存在以 TGF-β、PD-L1 与 MIF 为特征的免疫抑制性通讯重编程：TGF-β 驱动肿瘤 EMT 与免疫逃逸，PD-L1/PD-1 直接抑制 T 细胞杀伤，MIF 招募并极化免疫抑制细胞——三者协同将侵袭前沿塑造为 “免疫抑制 + 侵袭” 的枢纽（图 13-3）。

注意这段文字的两处纪律：一是每个 “提示” 背后都对应一张图与一个统计量；二是结论句用的是 “提示” 而非 “证明”——第 8 节的注意事项就是为这句话兜底。

8 注意事项

1. **差异≠因果**。生态位间的通讯差异是相关性发现：可能是通讯驱动了生态位（肿瘤重编程微环境），也可能是生态位环境（缺氧、ECM）驱动了通讯，还可能两者互相强化。要谈因果，需要扰动实验（中和抗体、敲除、类器官共培养）或至少时序数据。
2. **多生态位比较要做统计校正**。我们进行了成百上千次比较（通路 × 生态位、L-R 对 × 生态位），多重检验会膨胀假阳性率，默认使用 BH-FDR（Benjamini-Hochberg）校正，并在方法部分写明校正方法与阈值。
3. **重复样本是底线**。单个切片只能支持 “探索性假说”；任何 “重编程” 结论都应以多个生物学重复（多个患者/多个切片）为准，最好在独立队列中验证。
4. **组成差异的混淆**。生态位间通讯差异可能只是细胞组成差异的投影（CAF 多的地方当然 CAF 信号强），不一定是 “通讯程序被重编程”。严谨做法是结合去卷积丰度做组成匹配或校正，并在讨论中说明这一局限。
5. **分辨率与混合信号**。spot 的混合信号会让 “发送者/接收者” 标签带噪声；高分辨率平台（Visium HD、Xenium）或单细胞-空间联合分析可以缓解。

本章小结

本章完成了从 “差异” 到 “重编程” 的升级：用数量、强度、通路活性三个层面操作化定义重编程；用 compareInteractions / 手动汇总比较差异；用通路活性热图与 netVisual_diff 识别差异通路；用 log2FC 讲清差异 L-R 对的方向；用 Squidpy + COMMOT + 表达证据建立三重证据闭环；最后用 Sankey 图把故事讲完整。至此，全书分析主线（数据→生态位→通讯→重编程）已经跑通，下一章进入论文级图表与写作环节。

练习与思考

1. “通讯重编程” 的操作化定义包含哪三个层面？为什么说它是 “系统级差异” 而不是单点差异？
2. （动手题）用 `cc_list` 计算侵袭前沿 vs 肿瘤核心的差异 L-R 对（log2FC），列出上调最强的 5 对，并查它们在 CellChatDB 中属于哪些通路。
3. （动手题）在侵袭前沿内运行 COMMOT（dis_thr = 250）并绘制 TGFb 空间活性，与 CellChat 的 TGFb 网络结论对照，写一段 “三重证据” 判读。
4. （思考题）如果侵袭前沿的 CAF 比例远高于肿瘤核心，TGFb 的差异还能算 “重编程” 证据吗？你会怎么排除组成混淆？
5. （思考题）为什么单切片数据的 “重编程” 结论不能直接写进论文的结论部分？需要补什么？

第 14 章论文级图表与可视化

前置：已完成第 08–13 章的分析，手上有生态位、通讯、差异通讯的结果对象。本章教你把这些结果画成“能进论文”的图。

本章目标

1. 掌握论文级图表的通用规范：字号、配色、坐标轴、图注、分辨率与导出格式。
2. 学会 6 类关键图表的“何时用、怎么读、代码要点”：空间散点图、气泡图、通讯网络图、热图、堆叠条形图、Sankey 流量图。
3. 学会“一图一故事”的结果图编排策略，看懂 Figure 1–4 的组织逻辑，并以 `fig17_paper_figure.png` 为范例拆解 A/B/C/D 面板。
4. 掌握 R（showtext）与 Python（matplotlib 字体注册）的中文字体设置，解决中文乱码。
5. 能对照“常见图表错误表”自查并修正自己的图。

14.1 论文级图表通用规范

新手常犯的误区是“图能出就行”。论文审稿人看图的顺序是：先看整体是否整洁、再看字号是否可读、最后才看内容。下面 6 条规范是底线，请逐条对照自己的图。

（1）字号：正文 8–10 pt，图内最小不小于 6 pt。期刊排版会把图缩到栏宽（约 8–9 cm），图内文字如果小于 6 pt，印刷后几乎不可读。建议：图内文字 7–8 pt，轴标题与图例标题 9 pt。在 R 里用 `base_size` 统一控制，在 Python 里用 `rcParams` 统一控制。

（2）配色：色盲友好。全球约 8% 男性和 0.5% 女性有红绿色觉缺陷，红绿对比对他们是灾难。两个安全方案：

- 连续色（数值渐变）用 **viridis**：从深紫到亮黄，色觉缺陷下依然可区分明度变化。
- 分类色（离散组别）用 **Okabe-Ito** 8 色调色板：`#E69F00` 橙、`#56B4E9` 蓝、`#009E73` 绿、`#F0E442` 黄、`#0072B2` 深蓝、`#D55E00` 红、`#CC79A7` 粉、`#000000` 黑。本书教学图默认采用这一约定（各图配色见 `00-FIGURES` 清单）。

本书生态位图的四色约定以第 10 章 `fig10` 为准：肿瘤核心、侵袭前沿、免疫浸润区、基质区四种生态位使用四种可区分的分类色，全书统一，避免同一生态位在不同图里颜色不同。

（3）坐标轴。轴标题写清楚“变量（单位）”；刻度线朝外；刻度标签字号不小于 6 pt；坐标范围必须覆盖全部数据点，不要截断造成误导；两幅并列的图使用相同坐标范围，方便对比。

（4）图注要素（figure caption）。一句“结论性”标题 + 关键信息，通常包含：样本量 n、统计方法与校正（如“BH-FDR 校正，$\mathbf{P} < 0.05$”）、色标含义、必要的缩写解释。本书所有模拟数据图注必须注明“（模拟数据演示）”。

（5）分辨率：300 dpi。印刷需要 300 dpi，屏幕预览 150 dpi 足够。导出时直接设 300 dpi，宁可文件大一点。

（6）导出格式：PNG 用于投稿系统与正文预览，PDF（矢量）用于审稿与后续编辑。矢量图放大不糊，且文字可选，方便编辑替换字体。

下面是一段可运行的 R 导出模板（假设你在 RStudio 里）：

```r
# 目的：统一设置 ggsave 导出参数，保证全书图风格一致
library(ggplot2)

# ggplot 默认主题微调：图内文字 8 pt、白色背景、细网格
theme_paper <- theme_classic(base_size = 8) +
  theme(axis.ticks.length = unit(1.5, "pt"),
        legend.key.size = unit(3, "mm"))

p <- ggplot(mtcars, aes(mpg, wt)) + geom_point() + theme_paper

# PNG：300 dpi；PDF：矢量导出
ggsave("figures/figXX_example.png", p,
       width = 9, height = 8, units = "cm", dpi = 300)
ggsave("figures/figXX_example.pdf", p, width = 9, height = 8, units = "cm")
```

运行结果解读：`figures/` 下出现同名 PNG 与 PDF 两个文件，PNG 的像素尺寸约为 `9/2.54*300 × 8/2.54*300`，即约 1063×945 像素，满足印刷要求。

Python 侧对应设置：

```python
import matplotlib.pyplot as plt

plt.rcParams.update({
    "font.size": 8,                    # 全局字号 8 pt
    "axes.titlesize": 9,
    "axes.labelsize": 8,
    "xtick.labelsize": 7,
    "ytick.labelsize": 7,
    "figure.dpi": 300,                 # 输出分辨率
    "savefig.dpi": 300,
    "axes.edgecolor": "black",
```

```
})
fig, ax = plt.subplots(figsize=(3.5, 3.2))   # 约 9×8 cm
ax.scatter(range(10), range(10), s=12, c="#0072B2")
fig.savefig("figures/figXX_example.png", bbox_inches="tight")
fig.savefig("figures/figXX_example.pdf", bbox_inches="tight")
```

常见坑：只导 PNG 不导 PDF；字号用默认值（R 默认 11 pt、matplotlib 默认 10 pt），被期刊缩小后糊成一团；用红绿配色表达“上调/下调”。

14.2 关键图表制作清单

本书分析主线会产出 6 类图。每类图都按“何时用 + 怎么读 + 代码要点”给出，请对照你自己的结果选择。

14.2.1 空间散点图（SpatialDimPlot / sc.pl.spatial_scatter）

- **何时用**：把任何“按 spot 的属性”（QC 指标、聚类、细胞类型、生态位、通讯活性）映射回组织切片，回答“这个属性在组织上长在哪”。
- **怎么读**：颜色即属性；同一生态位的 spot 是否连成片；边界是否清晰；异常 spot（如高线粒体）是否集中在特定区域。
- **代码要点**：R 用 Seurat 的 `SpatialDimPlot`；Python 用 `squidpy.pl.spatial_scatter`。

```
# 目的：把生态位标签画回组织切片（第 10 章产物）
library(Seurat)
SpatialDimPlot(obj, group.by = "niche", label = FALSE, label.size = 3,
               cols = c("#E69F00", "#56B4E9", "#009E73", "#F0E442")) +
  theme(legend.position = "bottom")
ggsave("figures/fig10_niche.png", width = 9, height = 8, units = "cm", dpi = 300)
```

```
import squidpy as sq
sq.pl.spatial_scatter(adata, color="niche", size=0.4, cmap=None,
                      figsize=(4, 4), dpi=300)
```

运行结果解读：切片的 spot 网格按生态位着色，四种颜色形成四个空间区域；如果颜色呈“椒盐状”（大量孤立单点），说明生态位边界不稳定，回第 10 章调 q 或平滑参数。

14.2.2 气泡图（netVisual_bubble）

- **何时用**：展示“配体-受体对 × 细胞类型对（或生态位对）”的通讯，一行能同时看到强度（气泡大小）与显著性（颜色）。
- **怎么读**：行是 L-R 对，列是发送细胞→接收细胞；气泡越大通讯概率越高，颜色越深越显著；一眼看出“谁在给谁发信号”。
- **代码要点**：`netVisual_bubble(cellchat, sources.use, targets.use, signaling)`，可用 `signaling` 过滤到目标通路。

```
# 目的：展示侵袭前沿与免疫浸润区之间特定通路的 L-R 通讯（模拟演示）
library(CellChat)
netVisual_bubble(cellchat, sources.use = c("Tumor", "CAF"),
                 targets.use = c("CD8 T", "Macrophage"),
                 signaling = c("TGFb", "MIF", "PD-L1"),
                 font.size = 8, font.size.title = 9)
```

运行结果解读：图中每个气泡对应一对“发送细胞类型→接收细胞类型”；若某个 L-R 对整行空白，说明该通讯在当前分组里不显著，不要硬解读。

14.2.3 圆形通讯网络图（netVisual_circle）

- **何时用**：展示整体通讯网络的拓扑结构，或按生态位 split 后并列比较各生态位的“通讯图景”（对应本书 fig12）。
- **怎么读**：节点是细胞类型/生态位，连线粗细代表通讯强度，箭头方向代表发送→接收；节点越大说明参与通讯越活跃。
- **代码要点**：`netVisual_circle(net, weight.scale = TRUE)`；多生态位并列用 `netVisual_circle` 配合 `split.by` 或对每个生态位分别成图后用 `patchwork` 拼接。

```
# 目的：三个生态位的通讯网络并列（模拟演示，图 12-1 同款）
library(patchwork)
p_list <- lapply(c("Tumor core", "Invasive front", "Immune region"), function(n) {
  netVisual_circle(net_list[[n]], weight.scale = TRUE,
                   title.name = n, label.edge = FALSE)
})
wrap_plots(p_list, ncol = 3)
ggsave("figures/fig12_comm_network.png",
       width = 20, height = 7, units = "cm", dpi = 300)
```

运行结果解读：对比三张子图，侵袭前沿的“肿瘤→免疫”边是否明显变粗、是否出现独特的免疫抑制边（如 PD-L1 相关）——这正是“重编程”叙事的图形起点。

14.2.4 热图（ComplexHeatmap / pheatmap / matplotlib）

- **何时用**：矩阵型结果，如“生态位 × 信号通路活性”（fig15）、“细胞类型 × 细胞类型共定位 z 分数”（fig11）、“cluster × marker 基因表达”。
- **怎么读**：行/列聚类揭示结构；颜色渐变（如蓝→白→红）表示数值高低；配合行注释与列注释（如生态位分组色带）可同时读三层信息。
- **代码要点**：R 推荐 `ComplexHeatmap`（注释能力强）；Python 用 `seaborn.clustermap` 或 `matplotlib`。

```r
# 目的：生态位 × 通路活性热图（模拟演示，图 15-1 同款）
library(ComplexHeatmap)
library(viridis)
Heatmap(pathway_mat,                # 行 = 通路，列 = 生态位
        name = "activity",
        col = viridis(100),
        row_names_gp = gpar(fontsize = 8),
        column_names_gp = gpar(fontsize = 8),
        show_row_dend = TRUE, show_column_dend = FALSE)
# 导出：见 14.4 的 PDF 嵌入示例，先设 showtext 再 pdf()
```

运行结果解读：若侵袭前沿一列在 TGFb、MIF、PD-L1 等行同时呈高值（亮色块成簇），即为“侵袭前沿免疫抑制性重编程”的直观证据。

14.2.5 堆叠条形图

- **何时用**：展示组成比例，如“各生态位内细胞类型占比”（fig09 右）、“各生态位通讯数量/强度对比”（fig13 可用堆叠条形展示通路构成）。
- **怎么读**：条的总高 = 总数（或 100%），分段长度 = 各类占比；注意比较的是“比例”还是“绝对数量”，图注必须写清。
- **代码要点**：R 用 `ggplot2` 的 `geom_bar(position = "fill")`（比例）或 `position = "stack"`（数量）。

```r
# 目的：各生态位细胞组成比例（模拟演示）
library(ggplot2)
ggplot(comp_df, aes(x = niche, y = fraction, fill = celltype)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +
  scale_fill_manual(values = c("#E69F00", "#56B4E9", "#009E73",
                               "#F0E442", "#0072B2", "#D55E00", "#CC79A7")) +
  theme_classic(base_size = 8) +
  labs(x = " 生态位", y = " 细胞类型占比") +
  theme(legend.position = "right")
ggsave("figures/fig09_niche_composition.png", width = 10, height = 7,
```

```
       units = "cm", dpi = 300)
```

运行结果解读：横轴四个生态位、纵轴 0–100%，每个条内各细胞类型分段；免疫浸润区应有更高的 T/B 占比，基质区应有更高的 CAF 占比。

14.2.6 Sankey 流量图

- **何时用：**展示“来源生态位→重编程信号通路→靶细胞类型”的信号流（fig16），把“哪里的信号、通过什么通路、到达什么细胞”讲成一条可读的流。
- **怎么读：**从左到右三列节点，连线的宽度正比于通讯活性；流越粗的路径越是“重编程主轴”。
- **代码要点：**R 用 `ggsankey` 包（`remotes::install_github("daattali/ggsankey")`），数据整理为“来源-目标-权重”长表；Python 可用 `plotly` 的 `sankey`。

```
# 目的：重编程信号流 Sankey（模拟演示，图 16-1 同款）
library(ggsankey)
# df 三列：source(来源生态位) / target(通路或靶细胞) / value(通讯活性)
ggplot(df, aes(x = stage, next_x = next_stage,
               node = node, next_node = next_node,
               fill = node, value = value, label = node)) +
  geom_sankey(flow_alpha = 0.5, node.color = "gray30") +
  geom_sankey_label(size = 2.5, color = "black", fill = "white") +
  scale_fill_viridis_d() +
  theme_void(base_size = 8) + theme(legend.position = "none")
ggsave("figures/fig16_reprogramming.png", width = 14, height = 8,
       units = "cm", dpi = 300)
```

运行结果解读：从“侵袭前沿”出发的流若显著粗于其他来源，且汇入“PD-L1 / TGF-β”通路后指向 CD8 T 细胞，这就是全书的中心故事：**侵袭前沿通过免疫抑制通路重编程了 CD8 T 的微环境信号**。

14.3 一图一故事：结果图编排策略

有了 6 类单图，下一步是把它们编排成论文的“结果图”。原则是：**每张主图回答一个问题，每个面板是答案的一个证据**。本书建议的编排策略如下（对应 outline 的 Figure 1–4）。

- **Figure 1 路线图：**全书/课题的技术路线图（fig01 风格），一张图交代“数据→生态位→通讯→重编程”的流程，审稿人 30 秒内看懂你的分析框架。
- **Figure 2 生态位与组成：**生态位空间图（fig10）+ 邻域共现富集热图（fig11）+ 各生态位细胞组成堆叠条形（fig09 右），回答“组织里有哪些生态位、它们由什么细胞构成”。

- **Figure 3 通讯差异**：分生态位网络图（fig12）+ 通讯数量/强度柱状（fig13）+ 差异通讯气泡图（fig14），回答“生态位之间的通讯到底哪里不一样”。
- **Figure 4 重编程通路**：通路活性热图（fig15）+ 重编程信号流 Sankey（fig16），回答“差异背后的通路主轴是什么、信号流向哪里”，是全篇的高潮。

下面以本书范例图 `fig17_paper_figure.png`（模拟数据演示）讲解单张主图的 A/B/C/D 面板组织逻辑：

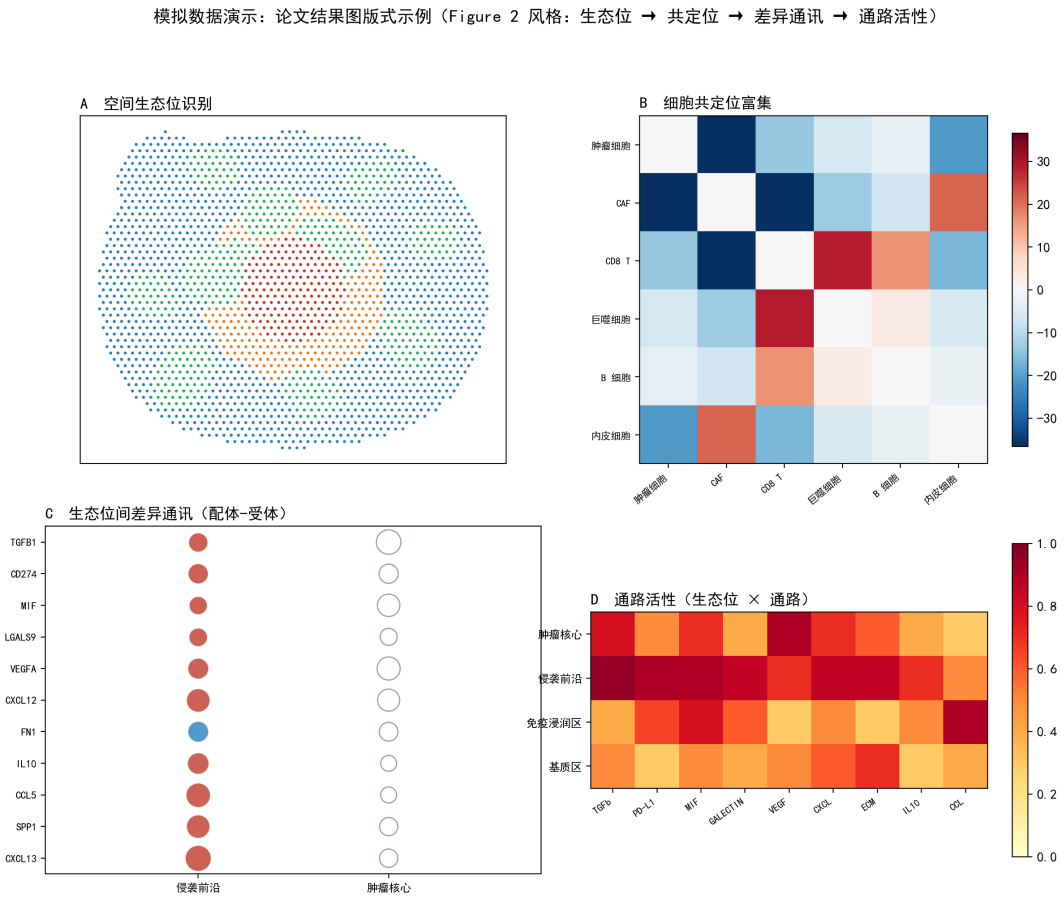

图 22: 图 14-1 论文结果图版式示例（模拟数据演示）。A：生态位识别（空间散点图，四色对应四种生态位）；B：邻域共现富集热图（细胞类型 × 细胞类型 z 分数，红 = 共定位富集）；C：差异通讯气泡图（侵袭前沿 vs 肿瘤核心，实心 = 前沿、空心 = 核心）；D：信号通路活性热图（生态位 × 通路）。

面板组织逻辑（阅读顺序：从左到右、从上到下，每面板回答一个问题）：

- **A 面板：先给空间背景。**“组织是什么样、生态位在哪”——用空间散点图建立空间直觉，是后续所有分析的空间锚点。
- **B 面板：给组成证据。**“生态位由什么细胞构成、哪些细胞喜欢待在一起”——用共现富集热图证明生态位不是拍脑袋分的，而是有细胞共定位依据的。
- **C 面板：给差异证据。**“生态位之间的通讯差在哪”——用差异气泡图把“重编程”落到具体的 L-R 对上。
- **D 面板：给机制整合。**“差异汇成哪些通路”——用通路热图把几十对 L-R 归纳成几条通路主轴，让故事可被一句话概括。

四面板合在一起回答完整问题：“在**哪里**（A），**谁和谁**在一起（B），**信号怎么变**（C），**归结为什么通路**（D）”。每个面板只承担一个任务，不要在一个面板里塞两件事。

图注写作模板：图 X 标题（结论句）。A：……（n=…，方法）；B：……（统计方法，色标含义）；C：……；D：……。模拟数据演示。

14.4 中文字体设置

R 的 ggplot 与 Python 的 matplotlib 默认字体都不含中文字形，直接输出中文会变成方块（乱码）。两个方案如下。

R：showtext 包（推荐，无需改系统字体）：

```r
# 目的：让 R 图正常显示中文，并嵌入 PDF
library(showtext)
font_add("SimHei", "simhei.ttf")        # Windows 自带黑体；Mac 可换 "PingFang SC"
showtext_auto()                         # 之后所有 ggplot 自动用 showtext 渲染
# 画图…然后导出：
pdf("figures/figXX_cn.pdf", width = 4, height = 3.5)
print(p)                                # p 为含中文的 ggplot 对象
dev.off()
showtext_end()
```

运行结果解读：`showtext_auto()` 之后，ggplot 中 `labs(x = "生态位")` 等中文即可正常渲染；`pdf()` + `dev.off()` 导出的 PDF 已内嵌字形，投稿系统打开不会缺字。

Python：matplotlib 字体注册：

```python
# 目的：注册中文字体并设为默认
import matplotlib
from matplotlib import font_manager

font_path = "C:/Windows/Fonts/simhei.ttf"   # Windows 黑体路径
font_manager.fontManager.addfont(font_path)
prop = font_manager.FontProperties(fname=font_path)
matplotlib.rcParams["font.family"] = prop.get_name()  # 设为默认字体
matplotlib.rcParams["axes.unicode_minus"] = False     # 修复负号显示为方块
```

运行结果解读：设置后，`ax.set_xlabel("生态位")` 等中文正常显示；`axes.unicode_minus = False` 是关键一行——不设置时负号会因字体回退而显示为方块。

常见坑：只注册不设默认（`rcParams["font.family"]`）；Linux 服务器上没有 SimHei（需先 `apt install fonts-wqy-zenhei` 或改用文泉驿/思源字体）；PDF 导出时未嵌入字体导致换机

后乱码（R 用 showtext、Python 用 `savefig` 的 `fonttype="cmap"` 可缓解，最稳妥是导出时确认字体内嵌）。

14.5 常见图表错误与修正

错误	后果	修正
图内文字小于 6 pt	印刷后不可读	统一 `base_size = 8` / `rcParams["font.size"] = 8`
红绿表达 “上调/下调”	色觉缺陷读者无法区分	改用蓝-黄或 viridis；或用形状/实心空心区分（见 fig14 的实心 = 前沿、空心 = 核心）
坐标轴范围截断	夸大差异、误导读者	坐标范围覆盖全部数据；对比图统一坐标
图注缺样本量与统计方法	审稿人要求补，返工	按 14.1(4) 模板补全
分辨率低于 300 dpi	印刷模糊	`dpi = 300`；矢量图用 PDF
中文乱码	图不可用	见 14.4 字体设置
一张面板塞两个问题	读者抓不住重点	拆成两张图，或改为一图一故事
两图并列但坐标/配色不一致	对比误导	并列图统一坐标范围、统一配色方案
忘了 “模拟数据演示” 标注	教学图被误当真实结果	图注统一加注（见 00-FIGURES）
只出 PNG 不出 PDF	编辑无法编辑文字	同时导出 PNG 与 PDF

本章小结

本章把 “分析结果” 变成 “论文级图表”。记住三个关键词：**规范**（字号 8–10 pt、色盲友好配色、300 dpi、图注四要素）、**清单**（6 类图各司其职：空间散点图给空间背景、气泡图给 L-R 细节、网络图给拓扑、热图给矩阵结构、堆叠条形给组成、Sankey 给信号流）、**故事**（Figure 1 路线→ Figure 2 生态位→ Figure 3 通讯差异→ Figure 4 重编程通路，每面板回答一个问题）。中文字体问题用 showtext / matplotlib 字体注册解决。最后对照 14.5 的错误表自查一遍，你的图就达到投稿门槛了。

练习与思考

1.（概念题）为什么论文图要用 300 dpi 导出？屏幕截图直接贴进稿件会有什么问题？
2.（动手题）打开第 10 章的生态位对象，用 `SpatialDimPlot` 把生态位画回切片，按 14.2.1 的规范调整字号与配色，导出 300 dpi PNG 与 PDF 各一张。
3.（动手题）用第 13 章的差异通讯结果画一张 `netVisual_bubble`，只保留 `signaling = c("TGFb", "PD-L1", "MIF")` 的 L-R 对，并写出它的图注（含统计方法与“模拟数据演示”标注）。
4.（思考题）对照 `fig17_paper_figure.png`，说明为什么 B 面板（共现富集）放在 C 面板（差异通讯）之前更合理。
5.（挑战题）把第 13 章的通讯结果整理成 Sankey 长表（来源生态位→通路→靶细胞），用 `ggsankey` 重画 fig16 同款图，比较它和热图在“讲故事”上的差别。

第 15 章结果解读与论文写作

前置：分析全部完成（第 08–13 章），图表已按第 14 章规范制作。本章把 “跑完的代码” 变成 “讲得通的故事”。

本章目标

1. 学会把分析结果组织成 “现象→机制→意义” 的重编程故事线。
2. 掌握结果章节 “四段式” 结构，拿到每个段落的模板与示例句。
3. 学会图表-文字对应与数字一致性检查，杜绝 “文字和图对不上” 的硬伤。
4. 掌握方法章节的可复现写作标准（数据来源、版本、参数）。
5. 了解适合空间组学/肿瘤方向的投稿期刊、bioRxiv 预印本流程与写作自查清单。

15.1 如何讲 “重编程” 故事

一篇结果论文的骨架是一句话能讲完的故事。本书课题的故事线是：

> **现象**：肿瘤组织内部并非均质，存在肿瘤核心、侵袭前沿、免疫浸润区、基质区等空间生态位，其细胞组成显著不同（fig10、fig11）。**机制**：不同生态位之间的细胞通讯网络系统性不同——侵袭前沿出现 TGF-β、PD-L1、MIF 等通路的异常激活（fig14、fig15）。**意义**：这种 “通讯重编程” 创造了免疫抑制微环境，可能解释免疫治疗响应差异，提示新的联合治疗靶点（需临床数据与实验验证）。

这条故事线的 “可视化终点” 就是第 13 章的通讯重编程流量图——一张图把 “来源生态位→信号通路→靶细胞” 的流向讲完，写作时它应出现在故事线的**机制**部分：

写作时的三条纪律：

- **现象与机制分开写**：先讲 “看到了什么”（描述性结果），再讲 “推测意味着什么”（解释性结果）。解释处一律用 “这提示……” “尚需实验验证” 等措辞，区分分析与推测（00-SPEC 第 8 节）。
- **机制要有证据链**：通讯差异（计算预测）必须配空间共定位证据（Squidpy 邻域富集、COMMOT 空间活性），否则只是 “表达相关”，站不住。
- **意义要克制**：没有生存分析或临床样本，就不要写 “预后价值”；最多写 “提示潜在的临床意义，值得后续研究”。

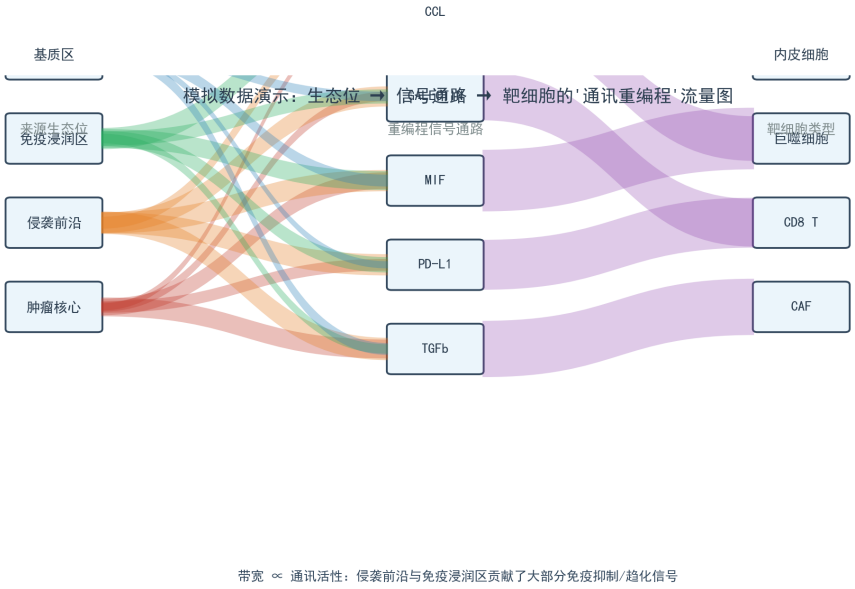

图 23: 图 15-1 通讯重编程信号流（模拟数据演示，同款见第 13 章 fig16）。带宽正比于通讯活性，展示“来源生态位→信号通路→靶细胞类型”的故事主线。

15.2 结果章节四段式

结果（Results）章节按“四段式”组织，每段回答一个问题。下面给出每段的**段落模板**与**示例句**，示例句中的数字均为模拟数据演示，写作时替换为你自己的真实输出。

15.2.1 第一段：生态位识别与验证

- **任务：**说明“我们如何得到生态位”，并给出验证证据（空间聚类 + 共定位富集 + 稳健性）。
- **段落模板：**我们用方法 X 对 Visium 切片进行空间聚类，识别出 N 个空间生态位。为验证其生物学意义，我们检查了各生态位的细胞组成与共定位模式。生态位边界在相邻切片/不同参数下保持稳定。
- **示例句：**利用 BayesSpace 对乳腺癌切片进行空间聚类（q = 4），我们识别出肿瘤核心、侵袭前沿、免疫浸润区与基质区四个空间生态位（图 2A）。Squidpy 邻域富集分析显示免疫细胞在免疫浸润区显著共定位（z > 3，图 2B），提示生态位划分与细胞空间组织一致。

15.2.2 第二段：各生态位特征

- **任务：**逐个生态位描述“由什么细胞构成、高表达什么 marker/通路”，用堆叠条形图与通路热图支撑。
- **段落模板：**生态位 X 以细胞类型 Y 为主（占比 p%，图 2C），高表达 marker Z 与通路 W。相比之下，生态位 X′……
- **示例句：**肿瘤核心以肿瘤细胞为主（占比 68%），高表达 EPCAM/KRT19 与 VEGF 通路；侵袭前

沿呈现肿瘤-基质交界特征，CAF 占比升高（31%），EMT 相关基因 VIM/ZEB1 与 TGF-β 通路活性显著高于肿瘤核心（图 2D，BH-FDR 校正，P < 0.05）。

15.2.3 第三段：生态位间通讯差异

- **任务**：报告“通讯的数量、强度、方向在生态位间如何不同”，这是“重编程”的操作化证据。
- **段落模板**：为刻画生态位特异的通讯程序，我们对每个生态位分别构建通讯网络（CellChat v2）。比较发现，生态位 X 的总通讯强度显著高于/低于……；差异 L-R 对主要富集于通路 A、B……
- **示例句**：分区通讯分析显示，侵袭前沿的总通讯强度约为肿瘤核心的 2.1 倍，且通讯方向由“肿瘤→免疫”转向“基质→肿瘤→免疫”（图 3A–B）。差异分析识别出 37 对在侵袭前沿显著上调的配体-受体互作（排列检验，BH-FDR 校正，P < 0.05，图 3C），包括 CXCL12–CXCR4、MIF–CD74 与 CD274–PDCD1。

15.2.4 第四段：重编程信号通路与空间证据

- **任务**：把几十对 L-R 归纳成通路主轴，并给出空间共定位证据，收束全篇。
- **段落模板**：上述差异互作可归纳为 N 条信号通路主轴。空间分析显示，配体细胞与受体细胞在侵袭前沿物理相邻（共定位富集/空间活性图），支持这些通讯在空间上真实发生。
- **示例句**：通路水平分析将差异互作归纳为 TGF-β、PD-L1、MIF 与 CXCL 四条主轴，其中免疫抑制相关通路（TGF-β、PD-L1）仅在侵袭前沿显著激活（图 4A）。空间验证显示，表达 CD274 的肿瘤细胞与表达 PDCD1 的 CD8 T 细胞在侵袭前沿显著共定位（邻域富集 z = 4.2；COMMOT 空间活性，距离阈值 250 μ m），为上述通讯推断提供了空间证据（图 4B）。这提示侵袭前沿存在免疫抑制性通讯重编程，可能削弱局部抗肿瘤免疫。

每段的字数建议：四段大约 250–400 词/段，加一段简短的开头（研究问题重述）与结尾（一句话总结）。

15.3 图表-文字对应与数字一致性检查

审稿人最常抓的问题：**正文数字与图表对不上**。下面的检查法能提前拦下这类错误：

1. **一句话一图原则**：正文每句“结果性陈述”必须能指出它来自哪张图/表（“图 2A”“表 S3”），没有出处的数字不写进结果。
2. **数字从输出抄录**：聚类数、占比、P 值一律从代码输出复制，不要凭记忆填写；差异分析结果统一从 `FindAllMarkers` / `compareInteractions` 的输出表导出，不要手抄。
3. **反向核对**：写完文字后，从每张图倒推一遍——图里出现的每个分组、每个数值，正文是否都提到了？图里有而正文没有的“孤儿信息”要么补写，要么删图。
4. **单位与样本量一致**：同一指标全文用同一单位（“通讯强度”与“通讯概率”不混用）；n

（spot 数/样本数）在正文、图注、方法三处必须一致。

5. **显著性表述规范**：写 “P $<$ 0.05（BH-FDR 校正）”，不要写 “显著（*）” 而不说明方法；效应量（如倍数变化）与显著性要分开报告。

一个实操技巧：建一张 “图表-文字对应表”，三列分别是 “正文陈述 / 出处图 / 输出文件与行号”，写完对照检查，比纯靠眼睛可靠得多。

15.4 方法章节写作：可复现是硬标准

方法（Methods）章节的目标是 “别人按你的描述能重跑出同样的结果”。三个必写要素：**数据来源、软件版本、参数**。检查标准是：每个分析步骤都能对应到一段可运行的代码。

方法章节推荐按分析流程分小节，与正文结果顺序一致：

```
方法章节骨架（伪代码式提纲）：
1. 数据来源：样本、平台、版本（如 10x Visium 人乳腺癌 demo，公开下载，见数据可得性声明）
2. 预处理与质控：过滤阈值（nFeature、nCount、percent.mt）及设定依据
3. 降维聚类与注释：SCTransform、PCA 前 30 维、resolution = 0.6、marker 注释/去卷积方法
4. 生态位识别：BayesSpace（q = 4、n.PCs = 15）与 Squidpy 邻域富集参数
5. 通讯分析：CellChat v2 默认流程 + computeCommunProb(type = "triMean")、排列检验次数
6. 差异通讯：生态位分区比较流程、统计与多重检验校正（BH-FDR）
7. 空间验证：COMMOT 距离阈值、Squidpy 共定位 z 分数
8. 统计与软件：R/Python 版本、包版本号、随机种子
```

写作要点：

- **版本写全**：`R 4.3.2`、`Seurat 5.0.1`、`CellChat v2`（`jinworks/CellChat` 2024 版）、`scanpy 1.10`、`squidpy 1.4`；写 “最新版” 等于没写。
- **参数写全**：阈值、维度、聚类数、随机种子（如 `set.seed(42)`）一个不能少。
- **数据可得性**：写明数据下载地址与版本（如 10x 官网数据集页的 demo 链接）；自己的数据要写 GEO 登录号（如有）。
- **代码可得性**：附 GitHub 仓库地址与 DOI（Zenodo 存档）；本书配套脚本（code/01–10）即为此而设计。
- **通信推断要声明为计算预测**：方法里写一句 “通讯分析为基于表达的计算预测，需实验验证”，既严谨又保护自己（00-SPEC 第 8 节）。

15.5 投稿期刊建议

以下仅为 “适合空间组学/肿瘤方向” 的选刊方向参考，**具体要求（格式、字数、费用）以投稿时各刊最新 Guide for Authors 为准，切勿照搬旧要求**：

- **Nature Communications**：接收空间组学方法与应用研究，影响大、开放获取。
- **Genome Medicine**：基因组/组学与临床结合方向，适合 “重编程与免疫治疗意义” 叙事。
- **Cell Press 肿瘤方向子刊**（如 Cancer Cell 及姊妹刊）：偏重生物学机制与临床转化，适合故事完整、有实验验证的工作。
- **Briefings in Bioinformatics**：偏方法综述/流程类，适合 “分析流程 + 教程式贡献”。
- 其他可选方向：空间组学专刊（如 Nature Methods 的 Methods 类工作——注意偏方法创新）、肿瘤免疫方向期刊（故事偏免疫微环境时）。

选刊三问：**①故事偏方法还是偏生物学？**方法创新投 Methods 系，生物学机制投肿瘤/免疫系；**②有没有实验验证？**纯计算工作投计算生物学友好期刊更稳妥；**③要不要开放获取？**影响选刊与经费。

15.6 预印本 bioRxiv 流程

投稿前先发预印本（preprint），好处是：抢占时间戳、免费获得社区反馈、投稿时可直接引用。流程：

1. **准备稿件**：主文 + 图 + 补充材料，整理为 PDF。
2. **登记作者与单位**：在 bioRxiv 注册账号，填写所有作者、单位、通讯作者邮箱。
3. **选择分类**：空间组学通常归 “Genomics” 或 “Bioinformatics” 分类（按网站当前分类选择最接近的）。
4. **上传并填写元数据**：标题、摘要、关键词、是否与已发表工作重叠声明。
5. **发布前检查**：确认所有图注、参考文献完整；确认没有未脱敏的临床信息。
6. **发布**：bioRxiv 会做基本筛查（非同行评审），通常 1–2 天内上线，获得 DOI。
7. **投稿后同步**：期刊投稿系统通常允许勾选 “已在 bioRxiv 发布”，无需撤回预印本；正式发表后预印本页面会自动关联期刊 DOI。

提醒：预印本不等同于同行评审，引用预印本时要说明 “预印本”；若后续大改，可发布修订版（revision）而不是另开一篇。

15.7 写作自查清单

提交前逐项打勾（可打印贴在屏幕旁）：

- ☐ 摘要一句话能复述 “现象→机制→意义” 故事线
- ☐ 结果四段式齐全，每段回答一个问题
- ☐ 每句结果性陈述都有图表出处（15.3 对应表已过）
- ☐ 数字全部来自代码输出，正文/图注/方法三处一致
- ☐ 解释性句子带 “这提示/推测”，区分分析结果与推测

- □ 统计方法（含 BH-FDR、排列检验）在正文与图注均有说明
- □ 通讯结论声明 “计算预测，需空间/实验验证”
- □ 方法含数据来源、软件版本、参数、随机种子
- □ 图符合第 14 章规范（字号/配色/300 dpi/图注）
- □ 参考文献格式统一（附录 B），预印本已注明
- □ 模拟数据图均已标注 “（模拟数据演示）”

本章小结

写作的本质是把 “分析输出” 翻译成 “科学论证”。本章给你的工具箱：**故事线**（现象→机制→意义）、**四段式结构**（生态位识别→特征→通讯差异→重编程通路，每段有模板与示例句）、**一致性检查**（一图一句、数字抄录、反向核对）、**可复现方法**（版本与参数写全）、**投稿路径**（选刊三问 + bioRxiv 流程）与**自查清单**。记住：审稿人读到的不是你的代码，而是你讲的故事——让每张图都服务于故事的一环。

练习与思考

1. （概念题）为什么 “通讯差异” 必须配空间共定位证据才能写进结果？只写计算预测会有什么风险？
2. （动手题）用第 13 章的输出（差异通讯表）写第三段 “生态位间通讯差异”，不少于 150 字，并标注每个数字的出处文件。
3. （动手题）把你自己分析中的一个数字写进 “图表-文字对应表”（正文陈述/出处图/输出文件行号三列），完成 3 行。
4. （思考题）你的课题如果完全没有临床数据，“意义” 段应该怎么写才不失严谨？写一个 50 字版本。
5. （挑战题）为你的课题起草 “方法章节骨架”（按 15.4 提纲 8 小节），并为每一步填上你实际使用的参数。

第 16 章常见问题与排错

前置：已完成或正在运行本书流程。本章是 “救火手册”：遇到报错先翻这里，按症状找对策。

本章目标

1. 能独立解决安装类问题：Rtools 缺失、Bioconductor 包编译失败、CellChat 依赖、Python 版本冲突、conda 环境混乱。
2. 掌握运行类问题的对策：内存不足、Seurat v5 与 v4 语法差异、cell2location 训练不收敛、BayesSpace 参数选择。
3. 解决可视化类问题：中文乱码、图像对齐、PDF 字体嵌入。
4. 理解结果类问题的根源：聚类不稳定、注释不准、通讯推断假阳性。
5. 会用交付前 10 项检查清单收尾。

16.1 安装类问题

安装问题占新手报错的 70%，且大多有固定解法。先记住一个总原则：**报错信息读最后 3–5 行**，通常真正的错误在末尾；前面几百行只是 “上下文”。

16.1.1 Rtools 缺失（Windows）

- **症状**：安装含 C/C++ 源码的 R 包时报 `compilation failed for package 'xxx'` 或 `ERROR: dependencies 'Rcpp', 'RcppArmadillo' are not available`，且找不到 `make`。
- **原因**：R 编译源码包需要 Rtools 工具链，RStudio 不会自动装。
- **对策**：到 CRAN 官网（`https://cran.r-project.org/bin/windows/Rtools/`）下载与你的 R 版本匹配的 Rtools 安装；安装后**重启 RStudio**，并确认 PATH：

```
# 目的：确认 Rtools 是否被 R 识别
Sys.which("make")
# 输出形如 C:\rtools43\usr\bin\make.exe 即为正常；若为空，说明 PATH 没生效
```

运行结果解读：若 `Sys.which("make")` 返回空字符串，先重启 RStudio；仍不行再手动把 `C:`

\rtools43\usr\bin 加入系统 PATH（安装时勾选 “Add to PATH” 可避免此问题）。

16.1.2 Bioconductor 包编译失败

- **症状**：BiocManager::install(”BayesSpace”) 报错，如缺依赖、编译失败。
- **原因**：Bioconductor 包依赖较多，且部分需要编译；版本不匹配（R 太旧或太新）。
- **对策**：先更新 BiocManager 并安装依赖，失败时看是 “缺哪个依赖” 再单独装：

```
# 目的：规范安装 Bioconductor 包
if (!requireNamespace(”BiocManager”, quietly = TRUE))
  install.packages(”BiocManager”)
BiocManager::install(version = ”3.19”)   # 与你的 R 版本匹配的 Bioc 版本
BiocManager::install(c(”SingleCellExperiment”, ”BayesSpace”),
                     update = TRUE, ask = FALSE)
```

运行结果解读：安装成功则 library(BayesSpace) 无报错。若仍失败，把报错里 ERROR: 后面第一行的包名记下来，单独 BiocManager::install(” 该包”) 逐个击破；Windows 上编译类失败优先检查 Rtools（16.1.1）。

16.1.3 CellChat 依赖（NMF / ComplexHeatmap）

- **症状**：devtools::install_github(”jinworks/CellChat”) 失败，或 library(CellChat) 报 there is no package called 'NMF'。
- **原因**：CellChat v2 依赖 NMF、ComplexHeatmap、presto 等包，未先装依赖时直接装 CellChat 会失败。
- **对策**：先装依赖再装 CellChat（NMF 与 ComplexHeatmap 优先）：

```
# 目的：先装 CellChat 的硬依赖，再装 CellChat 本体
install.packages(”NMF”)
BiocManager::install(”ComplexHeatmap”)
if (!requireNamespace(”remotes”, quietly = TRUE)) install.packages(”remotes”)
remotes::install_github(”jinworks/CellChat”)
library(CellChat)   # 不报错即成功
```

运行结果解读：library(CellChat) 后应能看到 CellChat 版本号（v2.x）。若 NMF 编译失败，回 16.1.1 检查 Rtools；ComplexHeatmap 失败则检查 BiocManager 版本。

16.1.4 Python 版本冲突

- **症状**：pip install scanpy 后 import scanpy 报 No module named，或装 A 包把 B 包版本升级坏了。

- **原因**：系统 Python 里混装多个项目的包；不同教程要求不同 Python 版本。
- **对策**：**永远用 conda 环境隔离**，不要在 base 环境乱装：

```
# 目的：创建隔离的 Python 环境（本书统一用 python=3.10）
conda create -n spatial python=3.10 -y
conda activate spatial
pip install scanpy squidpy anndata matplotlib pandas numpy
```

运行结果解读：`conda activate spatial` 后执行第 6 章冒烟测试里的 Python 版本检查命令，能打印版本号即环境可用。以后每次开新终端先 `conda activate spatial`，避免“装了很多但 import 不到”。

16.1.5 conda 环境混乱

- **症状**：`conda env list` 里一堆环境，不知哪个能用；或环境坏了想重来。
- **对策**：坏环境直接删掉重建，别修（省时间）：

```
conda env list                        # 看有哪些环境
conda env remove -n spatial -y        # 删除坏环境
conda create -n spatial python=3.10 -y    # 重建
# 好习惯：建好后立刻导出 yaml，日后一条命令复现
conda env export -n spatial > envs/spatial-niche.yaml
conda env create -f envs/spatial-niche.yaml    # 别人/换机复现
```

运行结果解读：yaml 文件记录完整环境（含版本），这是“可复现”的基石（详见第 15 章方法写作）；本书第 06 章有完整的环境搭建流程。

16.2 运行类问题

16.2.1 内存不足

- **症状**：R 报 `Error: cannot allocate vector of size ...` 或 Python 报 `MemoryError`；Windows 下 R 直接崩。
- **原因**：Visium 单切片约 5000 spot × 2 万基因，Seurat 运行时约需 2–6 GB；cell2location 训练建议 ≥ 16 GB（00-TECH 第 2 节）。
- **对策**（按性价比排序）：

```
# 对策 1：分析前只保留分析需要的细胞/基因，别贪多
obj <- subset(obj, subset = nFeature_Spatial > 200 &
                      nCount_Spatial > 500 & percent.mt < 25)
```

```r
# 对策 2：SCTransform 只对高变基因做（默认已如此），减少内存占用
obj <- SCTransform(obj, assay = "Spatial",
                   variable.features.n = 2000, verbose = FALSE)
# 对策 3：用完的大对象及时清掉
rm(large_obj); gc()
```

```python
# 对策 4（Python）：确认矩阵是稀疏格式，别转成稠密
import scipy.sparse as sp
print(sp.issparse(adata.X))          # 应为 True
# adata.X = sp.csr_matrix(adata.X)   # 若不是稀疏，转回稀疏
```

运行结果解读：过滤后对象变小，`gc()` 释放内存；稀疏矩阵对 0 多的表达矩阵省 10 倍以上内存。还不行就换大内存机器（或云上 32 GB 实例），这是最后手段。

16.2.2 Seurat v5 与 v4 语法差异

- **症状：**按网上的 v4 教程写代码报错，如 `CreateSeuratObject` 参数不认、`GetAssayData` 用法变了。
- **原因：**本书用 Seurat v5（00-SPEC 第 2 节），v5 重构了 assay 存储（分层系统），部分 v4 函数/参数不再兼容。
- **关键差异速查：**

操作	Seurat v4	Seurat v5
创建对象	`CreateSeuratObject(counts, assay = "RNA")`	空间数据用 `CreateSeuratObject(counts, assay = "Spatial")`
归一化	`NormalizeData()` / `SCTransform()`	`SCTransform(obj, assay = "Spatial")`（需指定 assay）
取表达矩阵	`GetAssayData(obj, slot = "data")`	`LayerData(obj, assay = "Spatial", layer = "data")` 或 `obj[["Spatial"]]$data`
读取空间数据	—	`Load10X_Spatial(data.dir, filename)`（v5 专用）
多 assay 操作	`obj@assays$RNA`	分层存储，`JoinLayers()` 后再操作旧式函数

对策：以本书 08 章代码为准；遇到 v4 教程的代码，先把“取数据”与“归一化”两处换成 v5 写法（上面表格），其他流程（PCA → FindNeighbors → FindClusters → UMAP）两版一致。

```r
# v5 正确写法（本书 08 章同款）
obj <- Load10X_Spatial(data.dir = "data/visium/",
                       filename = "filtered_feature_bc_matrix.h5")
obj <- SCTransform(obj, assay = "Spatial", verbose = FALSE)
```

运行结果解读：v5 下 `SCTransform` 必须给 `assay = "Spatial"`，漏了会报找不到默认 assay 的错误；`Load10X_Spatial` 是 v5 读取 Visium 数据的标准入口。

16.2.3 cell2location 训练不收敛

- **症状**：`model.train()` 的 ELBO 损失不下降或震荡；丰度估计全是背景噪声。
- **对策**（按顺序试）：

```python
# 目的：调整训练参数促进收敛（参数需按你的数据微调）
model.train(
    max_epochs=300,          # 默认可能偏少，加大
    batch_size=512,          # 显存允许时加大，稳定梯度
    lr=0.002,                # 学习率过大易震荡
    use_gpu=True,            # 有 GPU 强烈建议
)
```

运行结果解读：训练后查看 `model.history["elbo_validation"]` 曲线，应单调下降并平台化。若仍不收敛：①检查参考单细胞数据是否与 Visium 同组织类型、预处理是否一致；②先跑小步实验（subset 1000 个 spot）验证流程，再全量训练；③训练前 `model.view_summary()` 看参数是否合理。cell2location 训练建议内存≥ 16 GB（00-TECH 第 2 节）。

16.2.4 BayesSpace q 太大太慢

- **症状**：`spatialCluster(q = 15)` 跑几小时不出结果，或结果碎成椒盐状。
- **原因**：q 是聚类数。q 越大参数越多、MCMC 迭代越慢；q 与数据实际结构不匹配时结果也差。
- **对策**：q 从小到大试（4 → 7 → 10），对比 `clusterPlot` 结果选最合理的；先调低迭代数做快速预实验：

```r
# 目的：用小 q + 少迭代快速看趋势，再决定正式参数
library(BayesSpace)
sce <- spatialPreprocess(sce, platform = "Visium", n.PCs = 15)
sce <- spatialCluster(sce, q = 7, platform = "Visium", d = 15, nrep = 5000)
clusterPlot(sce)   # 快速查看 q=7 的分割效果
```

运行结果解读：`nrep` 控制 MCMC 迭代次数，调低只用于快速预览；正式分析建议默认（或按 10 章说明调高）。选择 q 的实用标准：空间连续、边界清晰、与先验结构（如肿瘤/免疫分区）吻合，而不是 q 越大越好。若单切片就要跑很久，检查是否同时开了太多后台任务（内存竞争）。

16.3 可视化类问题

16.3.1 中文乱码

- **症状：**图里中文变成方块 □□□。
- **原因：**默认字体不含中文字形。解法见第 14 章 14.4，这里给速查版：

```r
# R 速查（showtext）
library(showtext); font_add("SimHei", "simhei.ttf"); showtext_auto()
```

```python
# Python 速查（matplotlib 字体注册）
from matplotlib import font_manager
font_manager.fontManager.addfont("C:/Windows/Fonts/simhei.ttf")
import matplotlib.pyplot as plt
plt.rcParams["font.family"] = "SimHei"
plt.rcParams["axes.unicode_minus"] = False
```

运行结果解读：设置后重画即正常。若 `simhei.ttf` 找不到，先确认字体文件路径（Windows 一般在 `C:/Windows/Fonts/`）；Linux 服务器没有微软字体，改用 `fonts-wqy-zenhei`（文泉驿）或思源黑体。

16.3.2 图像对齐问题

- **症状：**`SpatialDimPlot` 里的点与组织 H&E 图像对不上（偏移、镜像、旋转 90°）。
- **原因：**图像坐标系与 spot 坐标不一致；加载时没正确读取 `scalefactors_json.json`，或图像被旋转过。
- **对策：**检查图像缩放因子与坐标读取是否完整：

```r
# 目的：核对空间数据的关键缩放信息（00-TECH 第 1 节）
json <- rjson::fromJSON(file = "data/visium/spatial/scalefactors_json.json")
json$tissue_hires_scalef; json$spot_diameter_fullres
```

运行结果解读:`tissue_hires_scalef` 是图像缩放到高分辨率图的因子,`spot_diameter_fullres` 是 spot 直径（像素）。如果 spot 大小明显不对，多半是缩放因子读错或用了错误的图像

文件（tissue_lowres_image.png 与 tissue_hires_image.png 对应不同缩放因子）。用 Load10X_Spatial 自动读取一般不会错；手写读取时务必用与图像匹配的因子。旋转/镜像问题多发生在自定义数据，检查你的坐标来源是否与图像同方向。

16.3.3 PDF 导出字体嵌入

- **症状**：PDF 在自己电脑正常，发给别人/投稿系统后中文或部分字符变方块。
- **原因**：PDF 未嵌入字体，对方机器没有对应字体时回退失败。
- **对策**：R 用 showtext 渲染后再导出（字体随 PDF 嵌入）；已有 PDF 可用 embedFonts 补救：

```
# 目的：给已有的 PDF 补嵌字体
showtext_auto()
pdf("figures/figXX_cn.pdf", width = 4, height = 3.5)
print(p)
dev.off()
embedFonts("figures/figXX_cn.pdf")   # 强制嵌入字体
```

运行结果解读：embedFonts 执行后 PDF 文件体积通常增大（字体被打包进去了），用 Adobe Reader 打开检查“字体”面板，应显示“已嵌入”。Python 侧在 savefig 时设 fonttype="cmap" 并保证字体已注册（16.3.1）即可。

16.4 结果类问题

16.4.1 聚类不稳定

- **症状**：换 resolution 或重跑一遍，聚类结果差很多；UMAP 图每次长得不一样。
- **原因**：Louvain/Leiden 聚类含随机性；分辨率决定聚类粒度。
- **对策**：固定随机种子 + 明确报告分辨率，多分辨率比较后选稳定方案：

```
set.seed(42)                                    # 固定随机种子，结果可复现
obj <- FindClusters(obj, resolution = 0.6)      # 记录所用 resolution
# 稳定性检查：跑 0.3 / 0.6 / 1.0 三个分辨率，选空间上最合理的一个
```

运行结果解读：set.seed 之后重复运行 FindClusters 得到相同结果；写论文时在方法里注明“resolution = 0.6，set.seed(42)”（第 15 章 15.4）。注意：固定种子只保证你自己的复现，换机器/换版本仍可能略有差异，属正常。

16.4.2 注释不准

- **症状**：去卷积/注释结果与先验不符（如免疫区域标出大量肿瘤细胞）；marker 打分注释明显错位。
- **原因**：marker 基因选择不当、参考单细胞数据不匹配、spot 混合信号难以区分相似细胞类型。
- **对策**：多策略交叉验证 + 空间一致性检查：

```
# 对策 1：换 marker 集合（本书 09 章统一 marker 见 00-TECH 第 7 节）
# 对策 2：去卷积换参考数据（cell2location/RCTD/SPOTlight 交叉验证）
# 对策 3：检查注释的空间一致性——同一生态位的 spot 注释应相近
```

运行结果解读：注释是“假设而非事实”。若两种方法结论冲突，优先信“空间上连续、生物学合理”的那个；把注释方法、参考数据、阈值写进方法章节，方便审稿人评估。

16.4.3 通讯推断假阳性：为什么必须空间验证

- **症状**：CellChat 报出大量显著互作，但其中很多在空间上“配体细胞和受体细胞根本不挨着”。
- **原因**：CellChat 的通讯概率只基于表达量 + 配体-受体先验库（质量作用定律 + 排列检验），**不利用空间位置**（00-TECH 第 5 节）。表达量高但物理上不相邻的两个细胞群，会被算成“有通讯”，这就是假阳性来源。下图是“计算上显著、但必须用空间证据复核”的典型产物：

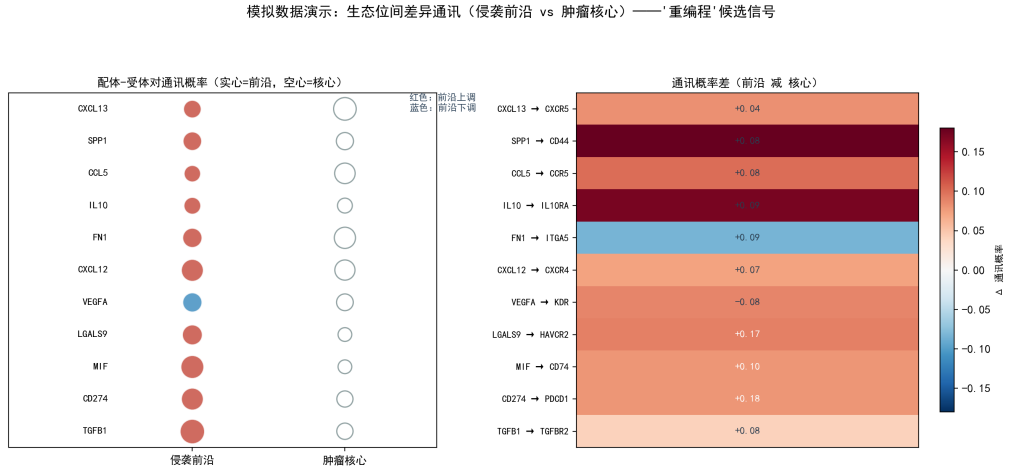

图 24: 图 16-1 生态位间差异通讯气泡图（模拟数据演示，同款见第 13 章 fig14）。红 = 侵袭前沿上调、蓝 = 下调；这类差异互作必须经空间共定位验证后才能写进结论。

- **对策**：三重验证，缺一不可：

```
# 验证 1：空间共定位（Squidpy 邻域富集，10 章已做）
sq.gr.nhood_enrichment(adata, cluster_key="celltype")
```

```
# 验证 2：空间活性（COMMOT，13 章，可选）
# import commot as ct; ct.tl.spatial_communication(adata, ...)
# 验证 3（最强）：实验——抗体中和、配体/受体敲除、荧光报告系统
```

运行结果解读：邻域富集 z 分数高的细胞对才支持“空间上真实发生通讯”；正文里写“通讯推断为计算预测，需空间共定位与实验验证”（00-SPEC 第 8 节）。**任何通讯结论必须配空间证据再写进论文**，这是本书反复强调的底线。

16.5 交付前检查清单

交稿/提交代码前，逐项打勾：

- ☐ 1. 环境可复现：`envs/spatial-niche.yaml` 已导出，换机 `conda env create -f` 能重建
- ☐ 2. 代码可运行：`code/01–10` 按顺序从头跑通一遍（至少验证关键脚本）
- ☐ 3. 随机种子已固定并在方法中说明（`set.seed(42)`）
- ☐ 4. 数据登记完整：来源、URL、md5、授权（data-inventory）
- ☐ 5. 图全部 300 dpi、PDF/PNG 双格式、图注含统计方法与样本量
- ☐ 6. 中文图无乱码，PDF 字体已嵌入
- ☐ 7. 正文数字与图表一致（15.3 对应表已过）
- ☐ 8. 统计方法（BH-FDR、排列检验）在正文与图注均已注明
- ☐ 9. 通讯结论均声明“计算预测”并附空间验证
- ☐ 10. 模拟数据图全部标注“（模拟数据演示）”

本章小结

排错的本质是“读报错→归因→最小化验证”。本章按四类问题给了速查表：**安装类**（Rtools、BiocManager、CellChat 依赖、conda 隔离）、**运行类**（内存、Seurat v5 语法、cell2location 收敛、BayesSpace 参数）、**可视化类**（中文、对齐、PDF 嵌入）、**结果类**（聚类、注释、通讯假阳性）。最后用 16.5 的 10 项检查清单收尾。遇到新报错时，记住：先读报错最后几行，再按“症状→原因→对策”定位，最后做最小实验验证修好了——这也是科学家解决问题的通用方法。

练习与思考

1.（概念题）为什么 `Sys.which("make")` 返回空字符串时，重装 R 包没有用？应该先做什么？
2.（动手题）在本机创建 `spatial` 环境并导出 `envs/spatial-niche.yaml`，然后删除环境、用 yaml 重建，验证可复现。

3.（动手题）故意把 Seurat v5 的 `SCTransform` 写成 v4 写法（不带 `assay`），运行并记录报错信息，再对照 16.2.2 修正。

4.（思考题）你的 CellChat 结果里有一对显著互作，但 Squidpy 邻域富集 z 分数接近 0。你会在论文里怎么写它？

5.（挑战题）对照 16.5 的 10 项检查清单，逐项检查你当前的交付物，写出每一项的通过/未通过状态与修复计划。

附录 A 术语表

全书术语中英对照速查。正文中每个术语首次出现时都有完整解释，本表供复习与写作时统一用词。排序说明：英文词条按字母序，中文词条按拼音序，两类混排在同一字母组内（英文在前、中文在后）。表格中的 “拼音首字母” 分组仅为排序用途。

A–D

- **AnnData**：scanpy 生态的核心数据结构，含表达矩阵 `X`、观察注释 `obs`、基因注释 `var`、降维结果 `obsm` 与非结构化信息 `uns`（见第 03 章）。
- **barcode（条形码）**：每个 spot 或细胞特有的核酸标签序列，用于把测序读段归回来源 spot/细胞（见第 02 章）。
- **BayesSpace**：基于贝叶斯模型的空间聚类方法，利用 spot 邻接信息把表达相似的相邻 spot 归为同一空间域（见第 10 章）。
- **BH-FDR**：Benjamini–Hochberg 校正后的错误发现率（false discovery rate），多重检验校正的默认方法，本书差异分析统一使用（见第 08/13 章）。
- **表达矩阵（expression matrix）**：行 × 列 = 基因 × spot（或细胞）的计数/表达量矩阵，空间分析的一切起点。
- **CAF（癌症相关成纤维细胞，cancer-associated fibroblast）**：肿瘤基质中的成纤维细胞，分泌 ECM 与生长因子，与侵袭前沿密切相关（见第 01 章）。
- **CellChat**：基于配体-受体库推断细胞通讯的 R 包，本书通讯分析的主工具，v2 支持空间模式（见第 11 章）。
- **CellChatDB**：CellChat 内置的配体-受体互作数据库，含分泌信号、ECM-受体、细胞-细胞接触三类（见第 11 章）。
- **cell2location**：基于深度生成模型的去卷积工具，用参考单细胞图谱估计每个 spot 上的细胞类型丰度（见第 09 章）。
- **CODEX**：基于抗体成像的空间蛋白组技术，可在组织原位同时检测数十种蛋白（见第 02 章）。
- **COMMOT**：基于集体最优传输（collective optimal transport）的空间通讯推断方法，输出配体-受体对的空间活性图（见第 13 章）。
- **参考单细胞图谱（reference scRNA-seq atlas）**：用于去卷积/注释的已有单细胞数据，需与目标组织类型匹配（见第 09 章）。
- **dropout（丢失事件）**：某基因在应表达的细胞中未被检测到的现象，单细胞/空间数据

普遍存在，影响表达解读。

- **单细胞转录组测序（scRNA-seq）**：逐个细胞测量基因表达的技术，细胞被解离后丢失空间位置（见第 02 章）。
- **多重检验校正（multiple testing correction）**：同时检验成千上万个基因/互作时，为控制假阳性而对 P 值进行的调整，如 BH-FDR（见第 08 章）。

E–L

- **ECM（细胞外基质，extracellular matrix）**：组织中的胶原、蛋白聚糖等非细胞组分，基质区生态位的主要特征（见第 01 章）。
- **EMT（上皮-间质转化，epithelial–mesenchymal transition）**：上皮细胞获得间质特征的细胞状态转变，与侵袭前沿和肿瘤侵袭相关（见第 01/10 章）。
- **H&E（苏木精-伊红染色）**：病理学常规染色，Visium 的配套组织图像即为 H&E 图，用于空间对齐与病理对照（见第 02 章）。
- **高变基因（highly variable genes）**：在细胞/spot 间表达波动大的基因，下游降维聚类前用于挑选特征基因（见第 08 章）。
- **高通量测序（high-throughput sequencing，NGS）**：并行测定海量 DNA/RNA 序列的技术，空间转录组的数据来源（见第 02 章）。
- **共现（co-occurrence）**：两类细胞/两组 spot 在空间上同时出现（相邻）的程度，邻域富集分析的基础（见第 10 章）。
- **归一化（normalization）**：校正测序深度（文库大小）差异，使不同 spot/细胞的表达量可比，如 Seurat 的 LogNormalize/SCTransform（见第 08 章）。
- **基质区（stromal region）**：以 CAF 与 ECM 为主的生态位，高表达 COL1A1/DCN/PDGFRA 等（见第 10 章）。
- **假阳性（false positive）**：被判定为“显著/存在”但实际上不成立的结果，通讯推断需空间验证来抑制（见第 16 章）。
- **降维（dimensionality reduction）**：把上万维基因空间压缩为少数主成分/嵌入坐标的方法，如 PCA、UMAP（见第 08 章）。
- **聚类（clustering）**：把相似的 spot/细胞分组（cluster）的无监督方法，如 Louvain、Leiden（见第 08 章）。
- **空间生态位（spatial niche）**：组织内具有特定细胞组成与功能的局部空间区域，本书核心概念，与空间域可互换（见第 04/10 章）。
- **空间域（spatial domain）**：空间聚类得到的分区，与空间生态位同义，注意全书行文统一（见第 04 章）。
- **空间转录组（spatial transcriptomics）**：在保留组织空间位置的同时测量基因表达的技术家族（见第 02 章）。
- **空间自相关（spatial autocorrelation）**：相邻位置的值比随机预期更相似（或更不同）的性质，Moran’s I 是其常用度量（见第 04 章）。

- **Leiden**：图聚类算法，Louvain 的改进版，scanpy 默认聚类方法，可调分辨率控制分组粒度（见第 08 章）。
- **邻域（neighborhood）**：某个 spot 周围与之空间相邻的 spot 集合，生态位与邻域分析的基本单元（见第 04/10 章）。
- **邻域富集（neighborhood enrichment）**：统计两类细胞在空间上是否比随机更常共定位，Squidpy 的 `nhood_enrichment` 即此分析（见第 10 章）。
- **Louvain**：经典的模块度优化图聚类算法，常用于单细胞/空间聚类（见第 08 章）。

M–R

- **marker 打分（marker scoring）**：按一组 marker 基因表达给 spot/细胞打 “类型倾向分” 的注释方法，如 AddModuleScore（见第 09 章）。
- **marker 基因（marker gene）**：某细胞类型特征性高表达的基因，如 CD3D 标记 T 细胞、EPCAM 标记肿瘤细胞（见第 09 章）。
- **MERFISH**：基于多重荧光原位杂交（multiplexed error-robust FISH）的空间转录组技术，单细胞分辨率（见第 02 章）。
- **Misty**：多视图解释框架（explainable multiview），把 spot 表达分解为自身、邻域、旁区视图的贡献（见第 04 章）。
- **Moran’s I**：度量空间自相关的统计量，取值越接近 1 表示越强的空间聚集（见第 04 章）。
- **免疫检查点（immune checkpoint）**：调节 T 细胞激活的抑制性信号，如 PD-L1/PD-1、CTLA-4，肿瘤利用其逃避免疫（见第 01 章）。
- **免疫浸润区（immune-infiltrated region）**：免疫细胞（T/B/巨噬细胞）富集的生态位，高表达 CD3D/CD8A/MS4A1 等（见第 10 章）。
- **免疫抑制（immunosuppression）**：微环境削弱抗肿瘤免疫反应的状态，本书 “重编程” 故事的核心后果（见第 01/15 章）。
- **NicheNet**：用配体-受体与下游调控网络推断 “配体→靶基因” 调控关系的工具（见第 04 章）。

P–S

- **PCA（主成分分析，principal component analysis）**：线性降维方法，把高维表达压缩为互不相关的主成分，聚类前常用（见第 08 章）。
- **PD-L1**：程序性死亡配体 1（CD274），与 T 细胞上的 PD-1 结合抑制免疫，免疫检查点通路代表（见第 01/13 章）。
- **排列检验（permutation test）**：通过随机重排数据构造零分布来评估显著性的非参数方法，CellChat 检验互作显著性即用此法（见第 11 章）。
- **配体-受体（ligand–receptor，L-R）**：细胞通讯的分子对——发送细胞表达配体、接收

细胞表达受体，通讯推断的基本单元（见第 01 章）。

- **批次效应（batch effect）**：因实验批次而非生物学差异引入的系统性表达偏差，多切片/多样本分析需校正（见第 08 章）。
- **侵袭前沿（invasive front）**：肿瘤与基质交界处的生态位，EMT 与免疫抑制通讯富集，本书重编程分析的重点区域（见第 10 章）。
- **去卷积（deconvolution）**：从混合信号（spot）反推各细胞类型比例的计算方法，如 cell2location、RCTD、SPOTlight（见第 09 章）。
- **趋化因子（chemokine）**：引导免疫细胞迁移的小分子分泌蛋白，如 CXCL12、CCL5，介导细胞招募型通讯（见第 01 章）。
- **RCTD**：基于单细胞参考的 spot 去卷积方法，输出各 spot 的细胞类型比例与置信度（见第 09 章）。

S–Z

- **Sankey 图（Sankey diagram）**：流量图，用连线宽度表示数值大小，本书用于展示 “来源生态位→通路→靶细胞” 的重编程信号流（见第 13/14 章）。
- **Scanpy**：Python 的单细胞/空间数据分析框架，本书 Python 侧主工具（见第 03/08 章）。
- **Seurat**：R 的单细胞/空间数据分析框架，本书 R 侧主工具，v5 支持 Visium 数据（见第 03/08 章）。
- **SingleCellExperiment（SCE）**：Bioconductor 的组学数据容器，BayesSpace 等包的数据接口（见第 03/10 章）。
- **SpaGCN**：结合基因表达、空间坐标与组织图像的图卷积聚类方法（见第 04 章）。
- **sparse matrix（稀疏矩阵）**：只存储非零元素的矩阵格式，表达矩阵多为稀疏矩阵，大幅节省内存（见第 16 章）。
- **spot**：10x Visium 捕获区上的捕获点，直径约 55 μm、中心距 100 μm，每个 spot 覆盖约 1–10 个细胞，本书保留英文（见第 02 章）。
- **SPOTlight**：基于非负最小二乘的 spot 去卷积方法（见第 09 章）。
- **Squidpy**：Python 的空间组学分析框架，提供邻域富集、空间统计与可视化（见第 10 章）。
- **STAGATE**：基于图自编码器的空间聚类方法（见第 04 章）。
- **TGF-β（转化生长因子β）**：多功能细胞因子，肿瘤中常促进 EMT 与免疫抑制，侵袭前沿高活性通路（见第 01/13 章）。
- **TME（肿瘤微环境，tumor microenvironment）**：肿瘤细胞及其周围的免疫、基质、血管等成分构成的局部环境（见第 01 章）。
- **t-SNE**：非线性降维/可视化方法，UMAP 的常见替代，注意其不保留全局结构（见第 08 章）。
- **通讯概率（communication probability）**：CellChat 用质量作用定律模型计算的配体-受体互作强度估计，值越大通讯越强（见第 11 章）。

- **通讯重编程（communication reprogramming）**：不同空间生态位间通讯网络/通路的系统性差异，本书核心概念与卖点（见第 13 章）。
- **UMAP**：非线性流形降维方法，广泛用于单细胞/空间数据的二维可视化（见第 08 章）。
- **UMI（唯一分子标识符，unique molecular identifier）**：每条 mRNA 分子上附着的随机标签，用于去除 PCR 重复、定量转录本数（见第 02 章）。
- **Visium**：10x Genomics 的空间转录组平台，组织切片贴片、原位捕获 mRNA（见第 02 章）。
- **文库大小（library size）**：单个 spot/细胞捕获到的总 UMI 数（nCount），反映测序深度，归一化前需校正（见第 08 章）。
- **细胞通讯（cell–cell communication，CCC）**：细胞间通过配体-受体等分子进行的信号交流，本书分析主线（见第 01 章）。
- **信号通路（signaling pathway）**：一组协同发挥功能的分子互作链，如 TGF-β 通路、PD-L1/PD-1 通路，通讯分析常按通路聚合解读（见第 13 章）。
- **异质性（heterogeneity）**：样本内部细胞组成或状态的多样性，空间转录组的核心研究对象之一。
- **质控（quality control，QC）**：按 UMI 数、基因数、线粒体比例等指标过滤低质量 spot/细胞的步骤（见第 08 章）。
- **肿瘤核心（tumor core）**：以肿瘤细胞为主、血管内皮丰富的生态位，高表达 EPCAM/KRT 与 VEGF（见第 10 章）。

附录 B 资源与参考文献

本附录汇总全书用到的工具官网、新手教程与参考文献。正文引用格式为 [作者 年份]（如 [Jin 2024]），与本附录列表对应。注：网站与文档 URL 可能更新，若失效请在搜索引擎按 “工具名 + documentation” 检索最新地址。

使用建议：本附录不是一次性读完的清单，而是 “用到再查” 的工具箱。第 06 章装环境时对照 B.1 核对包来源；第 07 章下载数据时对照 B.4；写第 15 章方法时对照 B.3 补全引用。所有条目都以 “官方一手来源” 优先——教程可以看二手，但版本、参数与数据格式必须以官方文档为准，这与第 15 章 “可复现是硬标准” 的原则一致。

B.1 工具官网与文档清单

工具	用途	官方地址
CellChat	细胞通讯推断（R，v2 支持空间模式）	GitHub：https://github.com/jinworks/CellChat
10x Genomics	Visium 平台与公开数据集	官网：https://www.10xgenomics.com/；数据集页：https://www.10xgenomics.com/datasets
Seurat	单细胞/空间数据分析（R）	官网：https://satijalab.org/seurat/
scanpy	单细胞/空间数据分析（Python）	文档：https://scanpy.readthedocs.io/
squidpy	空间组学分析（Python）	文档：https://squidpy.readthedocs.io/
BayesSpace	空间聚类（R，Bioconductor）	Bioconductor 页 + GitHub：https://github.com/edward130603/BayesSpace

工具	用途	官方地址
cell2location	spot 去卷积（Python，可选）	GitHub：`https://github.com/BayraktarLab/cell2location`
spatialLIBD	人脑 Visium 公开数据与工具（R）	Bioconductor 页 + GitHub：`https://github.com/LieberInstitute/spatialLIBD`
COMMOT	空间通讯推断（Python，可选）	GitHub：`https://github.com/zcang/COMMOT`
NicheNet	配体→靶基因调控推断（R）	GitHub：`https://github.com/saeyslab/nichenetr`

B.2 新手友好教程资源

资源	内容	与本书的对应
10x 官方分析指南（Space Ranger 文档与流程示例）	Visium 数据格式、标准分析流程	第 02/07/08 章
Seurat 空间转录组 vignette（`spatial_vignette.html`）	用 Seurat v5 跑通 Visium 全流程	第 08–09 章
scanpy 官方教程（`scanpy-tutorials.readthedocs.io`）	Python 侧预处理、聚类、可视化	第 08 章、附录 C
squidpy 官方教程	邻域富集、空间统计与可视化	第 10 章
CellChat 官方 tutorial（仓库内 Rmd 文档）	CellChat v2 完整流程与可视化	第 11–13 章
BayesSpace 官方 vignette	空间聚类参数与实操	第 10 章
本书配套脚本与教学图（`code/`、`figures/`）	按章可复现的分析代码与示例图	全书

学习建议：遇到某个工具不会用时，先打开它的官方 vignette 按示例跑一遍，再回本书看“这一步在课题里为什么做”。官方示例用的是干净数据，你的数据会报各种错——第 16 章排错手册就是为此准备的。建议按“官方文档→本书对应章节→回到自己的数据”三步循环：第一步建立工具语感，第二步理解分析逻辑，第三步才是解决自己数据里的实际问题。学习顺序上，先精读 Seurat 与 scanpy 两套入门教程（它们覆盖面最广），再按需查阅 squidpy、CellChat、BayesSpace 的专项文档，避免一开始就被过多的工具细节淹没。

B.3 参考文献

以下为本书核心文献，正文中按 [作者年份] 引用（如 CellChat 见 [Jin 2024]）。更多条目见 literature/references.bib。

1. Jin S, et al. 2024. CellChat for systematic analysis of cell–cell communication from single-cell and spatially resolved transcriptomics. *Nature Genetics*.（CellChat v2，第 11–13 章主工具）
2. Cang Z, et al. 2023. Screening cell–cell communication in spatial transcriptomics via collective optimal transport. *Nature Methods*.（COMMOT，第 13 章空间活性验证）
3. Palla G, et al. 2022. Squidpy: a scalable framework for spatial omics analysis. *Nature Methods*.（Squidpy，第 10 章邻域富集）
4. Zhao E, et al. 2021. Spatial transcriptomics at subspot resolution with BayesSpace. *Nature Biotechnology*.（BayesSpace，第 10 章生态位识别）
5. Browaeys R, et al. 2020. NicheNet. *Nature Methods*.（NicheNet 方法学，第 04 章）
6. Kleshchevnikov V, et al. 2022. Cell2location. *Nature Biotechnology*.（cell2location，第 09 章去卷积）
7. Pelka K, et al. 2021. Spatially organized multicellular immune hubs in human colorectal cancer. *Cell*.（结直肠癌空间生态位范例，第 15 章写作参考）
8. Maynard KR, et al. 2021. Transcriptome-scale spatial gene expression in the human DLPFC. *Nature Neuroscience*.（DLPFC Visium 数据，第 07 章数据源 spatialLIBD）
9. Hu J, et al. 2021. SpaGCN. *Nature Methods*.（SpaGCN，第 04 章方法学）
10. Dong K, Zhang S. 2022. STAGATE. *Nature Communications*.（STAGATE，第 04 章方法学）
11. Tanevski J, et al. 2022. Misty (explainable multiview). *Genome Biology*.（Misty 框架，第 04 章方法学）

引用规范提醒：以上条目仅给出 “作者-年份-标题-期刊” 四要素；投稿或写综述时，请按目标期刊要求补充卷、期、页码与 DOI，并核对作者全名单（以各刊正式收录版本为准）。正文引用时用 [作者 年份]，多个文献并列用分号分隔，如 “生态位识别方法综述见 [Zhao 2021; Dong & Zhang 2022]”。

如何扩展文献库：随着课题推进，你还会引用更多文献。维护方法很简单：每读到一篇相关论文，就把 “作者-年份-标题-期刊 + 一句话要点 + 下载链接” 追加到 literature/references.bib（BibTeX 格式）与 literature/notes/ 笔记里；写作时从笔记里检索，而不是临时上网找。本书第 04 章的方法学总览与第 15 章的写作流程都用到了这套文献工作流，保持条目风格与本附录一致即可。引用他人结论时务必回到原文核对，不要把二手转述当成一手来源。

B.4 数据资源

- **10x Visium 人乳腺癌 demo**：本书主示例数据，10x 官网数据集页可下载 “filtered feature-barcode matrix (H5)” 与 `spatial/` 目录（坐标、缩放因子、H&E 图像）。公开可用，引用时注明来源（见第 07 章）。
- **spatialLIBD 人脑 DLPFC Visium 数据**：`spatialLIBD::fetch_data("visium_IFDLPFC")`，来自 `[Maynard 2021]`。
- **GEO**：检索 “Visium” 可找到大量公开空间数据，注意核对授权与引用要求（见第 07 章 data-inventory）。

数据获取的一般原则：优先使用公开、有明确版本与授权说明的数据（如 10x demo、spatialLIBD），便于复现与引用；下载后立刻记录样本 id、URL、md5 与授权信息（第 07 章 data-inventory 规范），这是第 15 章 “数据可得性声明” 的基础。涉及临床样本或受控访问的数据（如 dbGaP、EGA）需要申请权限，本书不涉及此类数据；若你的课题用到，请务必遵守数据使用协议，并在论文中如实声明。

附录 C 示例数据与代码清单

本附录是“复现全书”的总索引：10 个配套脚本按什么顺序跑、每个脚本干什么、示例数据从哪来。所有脚本位于 `code/`，教学图位于 `figures/`。

C.1 配套代码清单

全书配套 10 个 R 脚本（00-SPEC 第 3 节），按分析流程编号命名，与章节目录无关。**建议运行顺序就是编号顺序**：每个脚本的输出是下一个脚本的输入，请勿跳号运行。运行时长均为估计值，依机器配置（CPU/内存/网速）而异，仅供参考。

全书配套 10 个 R 脚本（00-SPEC 第 3 节），按分析流程编号命名，与章节目录无关。**建议运行顺序就是编号顺序**：每个脚本的输出是下一个脚本的输入，请勿跳号运行。运行时长均为估计值，依机器配置（CPU/内存/网速）而异，仅供参考。全部脚本位于 `code/` 目录：

1. **第 1 步 · `code/01_setup.R`（对应第 06 章，30–60 分钟）**：环境与包——安装/加载 Seurat v5、CellChat v2、BayesSpace 等并检查版本；
2. **第 2 步 · `code/02_download.R`（对应第 07 章，10–30 分钟）**：下载 10x Visium 人乳腺癌 demo（H5 + spatial 目录），校验 md5 并登记数据清单；
3. **第 3 步 · `code/03_preprocess.R`（对应第 08 章，10–20 分钟）**：预处理与质控——QC 指标、过滤、SCTransform、PCA、聚类、UMAP、空间可视化；
4. **第 4 步 · `code/04_annotation.R`（对应第 09 章，30–60 分钟）**：细胞类型注释——marker 打分 + cell2location 去卷积调用说明（Python 部分见同目录 `.py` 脚本）；
5. **第 5 步 · `code/05_spatial_domain.R`（对应第 10 章，30–60 分钟）**：生态位识别——BayesSpace 空间聚类 + Squidpy 邻域富集（Python 调用）+ 生态位命名与特征化；
6. **第 6 步 · `code/06_cellchat.R`（对应第 11 章，20–40 分钟）**：CellChat 基础——构建对象、通讯概率、通路聚合、经典可视化（气泡/网络）；
7. **第 7 步 · `code/07_niche_cellchat.R`（对应第 12 章，30–60 分钟）**：生态位分区通讯——按生态位 subset 后分别建 CellChat，比较生态位特异的通讯程序；
8. **第 8 步 · `code/08_differential_comm.R`（对应第 13 章，20–40 分钟）**：差异通讯/重编程——compareInteractions、netVisual_diff、通路对比 + COMMOT 空间活性（可选）；
9. **第 9 步 · `code/09_visualization.R`（对应第 14 章，10–20 分钟）**：论文级图表——统一字号/配色/300 dpi 导出；
10. **第 10 步 · `code/10_summary.R`（对应第 15 章，5–10 分钟）**：汇总输出——生成结果表

（CSV）、图表-文字对应表草稿、方法参数汇总。

运行总览（文字版流程图）：脚本间依赖与章节目对应如下，箭头表示 “前一产物是后一输入”：

```
02_download.R  ──►  03_preprocess.R  ──►  04_annotation.R  ──►  05_spatial_domain.R
(07 章 数据)       (08 章 预处理)         (09 章 注释)            (10 章 生态位)
      ▲
      │ 数据
      │
01_setup.R (06 章 环境, 先于一切)

05_spatial_domain.R ──► 06_cellchat.R ──► 07_niche_cellchat.R ──► 08_differential_comm.R
(生态位标签)            (11 章 CellChat)  (12 章 分区通讯)        (13 章 差异/重编程)

08_differential_comm.R ──► 09_visualization.R ──► 10_summary.R
(差异表/通路活性)          (14 章 论文级图表)       (15 章 汇总报告)
```

按章节对应关系再读一遍：**07 章数据（02 脚本）→ 08 章预处理产物（03 脚本）→ 09 章注释（04 脚本）→ 10 章生态位（05 脚本）→ 11 章 CellChat（06 脚本）→ 12 章分区通讯（07 脚本）→ 13 章差异/重编程（08 脚本）→ 14 章图表（09 脚本）→ 15 章写作汇总（10 脚本）**。跳过任何一环，下游脚本都会因缺输入报错。

C.2 示例数据说明

C.2.1 10x Visium 人乳腺癌 demo（主示例数据）

- **获取**：10x Genomics 官网数据集页（`https://www.10xgenomics.com/datasets`）搜索 “Human Breast Cancer”（Visium 系列），下载两项：① “filtered feature-barcode matrix (H5)”；② `spatial/` 目录（含 `tissue_positions_list.csv`、`scalefactors_json.json`、`tissue_hires_image.png`、`tissue_lowres_image.png`）。下载命令与校验见 `code/02_download.R` 与第 07 章。
- **结构**（00-TECH 第 1 节）：捕获区 6.5×6.5 mm、约 5000 个 spot（直径≈ 55 μm、中心距 100 μm）；H5 内含 `barcodes`（spot 条形码）、`features`（基因）、`matrix`（计数，稀疏格式）；`spatial/` 提供坐标、缩放因子与 H&E 图像。
- **为什么选它**：乳腺癌组织结构典型、四种生态位（肿瘤核心/侵袭前沿/免疫浸润区/基质区）分明，适合新手一眼看懂生态位概念（第 07 章）。
- **登记要求**：下载后按 data-inventory 规范登记样本 id、URL、md5 与授权（10x demo 为公开可用，引用注明来源）。

C.2.2 教学模拟数据 data/sim_visium.npz

- **用途：**教学图（fig07–fig17）的“数据分析结果”类图全部基于**模拟数据演示**（00-FIGURES 声明），让新手在**无需下载大文件**的情况下看到“真实分析会长什么样”。模拟数据体积小、生成快，适合课堂与快速试跑。
- **结构与读取：**`data/sim_visium.npz` 为 numpy 压缩格式，含模拟的 spot× 基因计数矩阵、坐标、生态位/细胞类型标签等键（键名以脚本内注释为准）。用 `numpy.load` 读取即可。

```
# 目的：查看模拟数据的键与形状（快速试跑用）
import numpy as np
d = np.load("data/sim_visium.npz")
print(list(d.keys()))          # 如 counts / coords / niche / celltype ...
print(d["counts"].shape)       # (n_spots, n_genes)
```

- **生成脚本：**位于 `code/figures/`，其中 `gen_figures_partA_schematics.py` 生成概念示意图（fig01–fig06），`gen_figures_partB_analysis.py` 生成模拟分析结果并产出 `data/sim_visium.npz` 与全部模拟结果图（fig07–fig17）。运行该脚本会重新生成模拟数据与教学图，可用于核对图注与数据的一致性。
- **重要提醒：**模拟数据只用于**学方法**，不可用于真实课题结论；真实课题请用 10x demo 或自己的数据（第 07 章）。

C.3 复现全书的最低路径

想最快看完全书流程长什么样，按此顺序：

1. 跑 `code/01_setup.R` 装好环境（首次约 1 小时）；
2. 跑 `code/02_download.R` 下载 demo 数据；
3. 依次跑 `03` → `04` → `05` → `06` → `07` → `08`（合计约 3–4 小时，cell2location 部分可选）；
4. 跑 `09_visualization.R` 得到论文级图表，对照 `figures/` 中的教学图检查；
5. 跑 `10_summary.R` 生成汇总表，作为第 15 章写作的素材。

每跑完一个脚本，对照该章“运行结果解读”检查输出是否符合预期；不符合就先排错（第 16 章）再继续，不要带着错误往下跑。